US010091287B2

(12) **United States Patent**
Larabie-Belanger

(10) **Patent No.: US 10,091,287 B2**
(45) **Date of Patent: Oct. 2, 2018**

(54) **DETERMINING PRESENCE IN AN APPLICATION ACCESSING SHARED AND SYNCHRONIZED CONTENT**

(71) Applicant: **Dropbox, Inc.**, San Francisco, CA (US)

(72) Inventor: **Maxime Larabie-Belanger**, San Francisco, CA (US)

(73) Assignee: **Dropbox, Inc.**, San Francisco, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 214 days.

(21) Appl. No.: **14/248,147**

(22) Filed: **Apr. 8, 2014**

(65) **Prior Publication Data**

US 2015/0288756 A1 Oct. 8, 2015

(51) **Int. Cl.**

G06F 3/048	(2013.01)
H04L 29/08	(2006.01)
H04L 29/06	(2006.01)
G06F 3/0484	(2013.01)
G06F 17/30	(2006.01)

(52) **U.S. Cl.**

CPC ***H04L 67/1095*** (2013.01); ***G06F 3/04842*** (2013.01); ***G06F 17/30165*** (2013.01); ***H04L 65/403*** (2013.01); ***H04L 67/24*** (2013.01)

(58) **Field of Classification Search**

CPC . H04L 67/1095; H04L 67/24; G06F 3/04842; G06F 17/30165

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,337,407	A	8/1994	Bates et al.	
5,960,173	A	9/1999	Tang et al.	
7,386,529	B2	6/2008	Kiessig et al.	
7,496,633	B2	2/2009	Szeto et al.	
7,610,280	B2	10/2009	O'Toole et al.	
7,631,007	B2	12/2009	Morris	
7,640,506	B2 *	12/2009	Pratley	G06Q 10/10 715/751
7,676,526	B1	3/2010	Beloussov et al.	
7,769,810	B1 *	8/2010	Kaufman	G06F 17/30165 709/205

(Continued)

OTHER PUBLICATIONS

Lim, H. et al., "Ubi-Jector," CHI 2013: Changing Perspectives, Apr. 27, 2013-May 2, 2013, pp. 1695-1700.

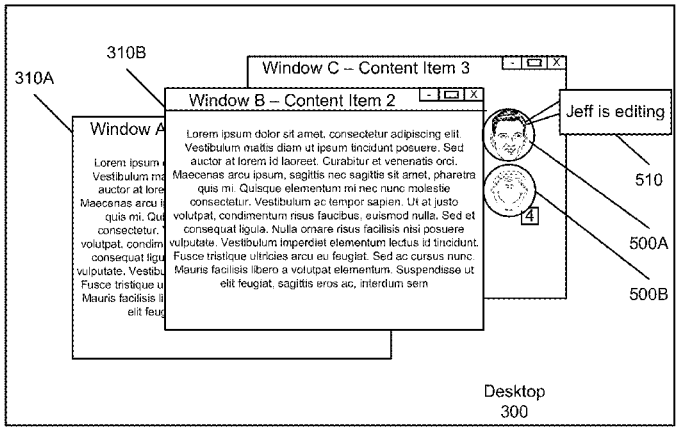
(Continued)

Primary Examiner — Andrew T Chiusano
(74) *Attorney, Agent, or Firm* — Fenwick & West LLP

(57) **ABSTRACT**

A device collects presence information and other interaction information from an application viewing a content item synchronized with a content management system. The interaction information indicates interactions of a device with respect to a content item, and includes presence information obtained from a native application such as whether the content item is being viewed by the user on a user interface element or the user interface element is modifying the content item. A presence management module receives presence events indicating possible change of presence with respect to a user interface window associated with a process and a synchronized content item. Such presence events include a change in focus of a user interface element indicating that a user is viewing the content item, and changes to a content item indicating a user is editing the content item.

23 Claims, 13 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

7,818,419	B1 *	10/2010	McAllister	G06Q 30/02 370/252
8,055,644	B2	11/2011	Crowley et al.	
8,244,830	B2	8/2012	Robinson et al.	
8,301,606	B2	10/2012	Futatsugi	
8,452,959	B2	5/2013	Brown et al.	
8,600,934	B2	12/2013	Strong et al.	
8,615,552	B2	12/2013	Shah	
8,655,840	B2	2/2014	Simelius	
8,700,759	B2 *	4/2014	Chakra	H04L 67/1095 709/223
8,825,597	B1	9/2014	Houston et al.	
8,868,500	B2	10/2014	Giampaolo et al.	
9,021,113	B2	4/2015	Carr et al.	
9,338,242	B1	5/2016	Suchland et al.	
9,773,010	B1	9/2017	Ganesh et al.	
9,773,051	B2	9/2017	Smith	
2003/0018719	A1 *	1/2003	Ruths	G06F 9/52 709/205
2004/0058710	A1	3/2004	Timmins et al.	
2006/0004911	A1	1/2006	Becker et al.	
2006/0101064	A1	5/2006	Strong et al.	
2006/0190117	A1	8/2006	Weczorek et al.	
2007/0174787	A1	7/2007	Rhee	
2008/0005235	A1 *	1/2008	Hegde	G06Q 10/10 709/204
2008/0109547	A1	5/2008	Bao et al.	
2008/0307102	A1	12/2008	Galloway et al.	
2009/0228504	A1	9/2009	Brown et al.	
2009/0271708	A1	10/2009	Peters et al.	
2009/0288150	A1	11/2009	Toomim et al.	
2010/0017879	A1 *	1/2010	Kuegler	G06F 21/123 726/23
2010/0131868	A1	5/2010	Chawla et al.	
2010/0138387	A1	6/2010	Simelius	
2010/0198874	A1	8/2010	Futatsugi	
2010/0257457	A1 *	10/2010	De Goes	G06Q 10/10 715/751
2011/0055177	A1 *	3/2011	Chakra	G06F 17/30165 707/695
2012/0101980	A1	4/2012	Taleghani et al.	
2012/0124486	A1	5/2012	Robinson et al.	
2012/0185434	A1	7/2012	Giampaolo et al.	
2012/0278127	A1	11/2012	Kirakosyan et al.	
2012/0303716	A1	11/2012	Peters et al.	
2012/0324121	A1	12/2012	Carr et al.	
2012/0331108	A1	12/2012	Ferdowsi et al.	
2013/0013560	A1 *	1/2013	Goldberg	G06F 17/30174 707/634
2013/0054509	A1 *	2/2013	Kass	G06Q 10/00 706/55
2013/0124638	A1	5/2013	Barreto	
2013/0124967	A1	5/2013	Hatfield et al.	
2013/0138608	A1	5/2013	Smith	
2013/0151596	A1	6/2013	Strong et al.	
2013/0185651	A1 *	7/2013	Wionzek	H04L 67/24 715/753
2013/0191339	A1 *	7/2013	Haden	G06F 17/3007 707/638
2013/0191451	A1 *	7/2013	Tse	G06Q 10/101 709/204
2013/0212250	A1 *	8/2013	Kleppner	G06F 17/30867 709/224
2013/0254656	A1 *	9/2013	Florian	G06F 17/2288 715/255
2013/0297680	A1 *	11/2013	Smith	H04L 29/08117 709/203
2014/0019626	A1	1/2014	Hubler et al.	
2014/0101310	A1 *	4/2014	Savage	H04W 4/003 709/224
2014/0156749	A1	6/2014	Kanter et al.	
2014/0172799	A1	6/2014	Dorman	
2014/0201289	A1	7/2014	Wheeler et al.	
2014/0223333	A1	8/2014	Pegg et al.	
2014/0237387	A1 *	8/2014	Ryall	H04L 65/403 715/753
2014/0244583	A1 *	8/2014	Wang	G06F 17/30368 707/625
2014/0279846	A1 *	9/2014	Srinivasan	G06F 17/30011 707/608
2014/0372852	A1 *	12/2014	Rothschiller	G06F 17/246 715/212
2015/0012488	A1 †	1/2015	van Rossum	
2015/0082196	A1 *	3/2015	Berger	G06Q 10/101 715/753
2015/0149404	A1 *	5/2015	Lock	H04L 65/403 707/608
2015/0261411	A1 *	9/2015	Snook	G06Q 10/101 715/751
2015/0278330	A1 *	10/2015	Hawa	G06F 17/30581 709/203
2015/0286655	A1 *	10/2015	von Haden	G06F 17/3007 707/610
2015/0288757	A1 *	10/2015	Larabie-Belanger	H04L 67/1095 709/206
2015/0288774	A1 *	10/2015	Larabie-Belanger	H04L 67/24 715/758
2015/0288775	A1 *	10/2015	Larabie-Belanger	H04L 67/24 715/203
2016/0119414	A1 *	4/2016	Westaway	G06Q 10/107 709/205
2016/0259508	A1 *	9/2016	Eccleston	G06F 3/0484

OTHER PUBLICATIONS

PCT International Search Report and Written Opinion, PCT Application No. PCT/IB2015/052522, dated Jun. 18, 2015, 11 pages.

PCT International Search Report and Written Opinion, PCT Application No. PCT/IB2015/052519, dated Jun. 18, 2015, 9 pages.

Adahl, J., "Shared Resource for Collaborative Editing Over a Wireless Network," Master of Science Thesis, University of Gothenburg, Dec. 2010, 274 pages.

Terry, D.B. et al., "Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System," Proceedings of the 15th ACM Symposium on Operating System Principles, Dec. 3-6, 1995, pp. 1-12.

PCT International Search Report and Written Opinion, PCT Application No. PCT/IB2015/052521, dated Sep. 24, 2015, 8 pages.

PCT International Search Report and Written Opinion, PCT Application No. PCT/IB2015/052523, dated Sep. 24, 2015, 7 pages.

European Examination Report, European Application No. 15736317.7, dated Oct. 30, 2017, 4 pages.

European Examination Report, European Application No. 15736316.9, dated Oct. 30, 2017, 5 pages.

* cited by examiner
† cited by third party

Fig. 1

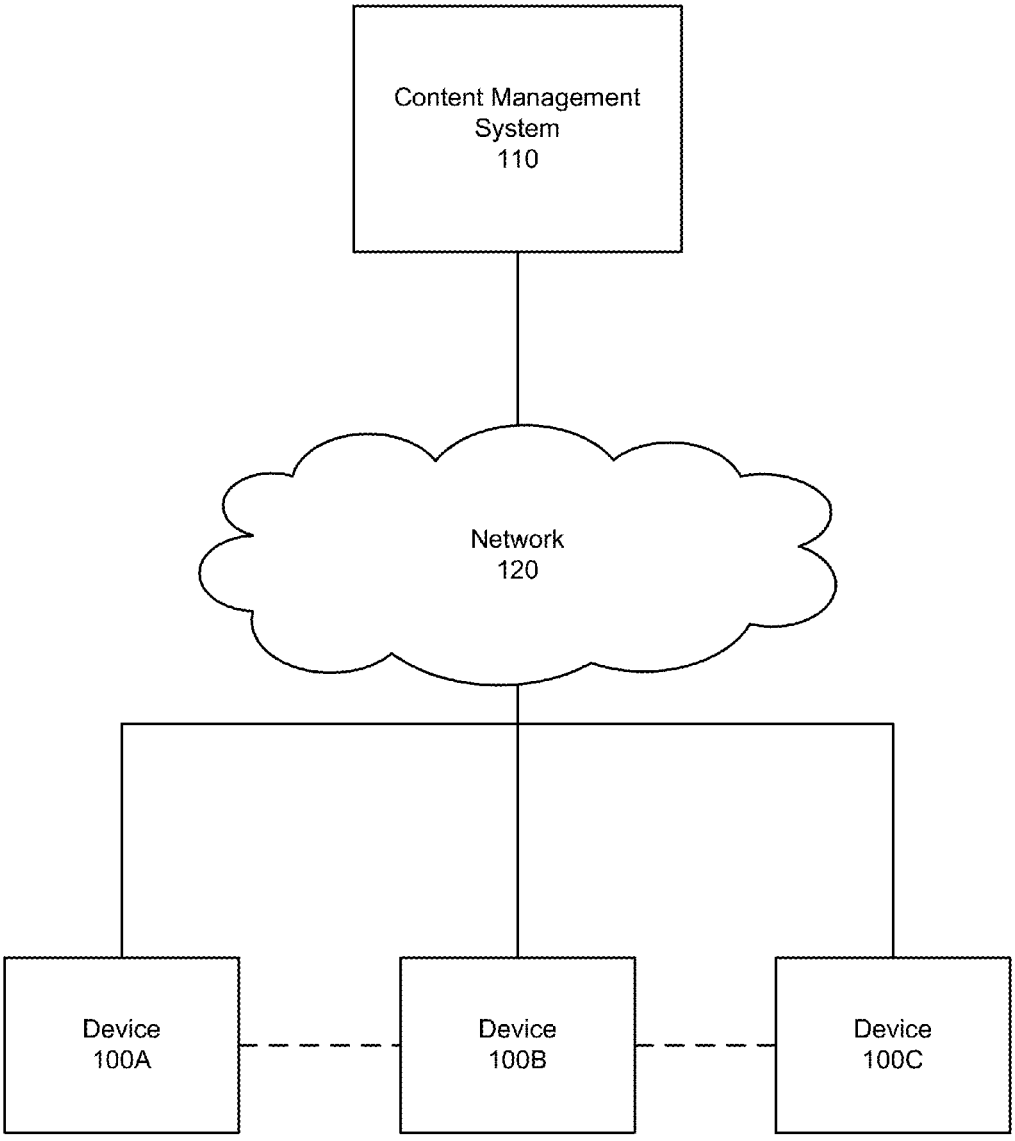

Content Management System 110
Network 120
Device 100A
Device 100B
Device 100C

Fig. 2

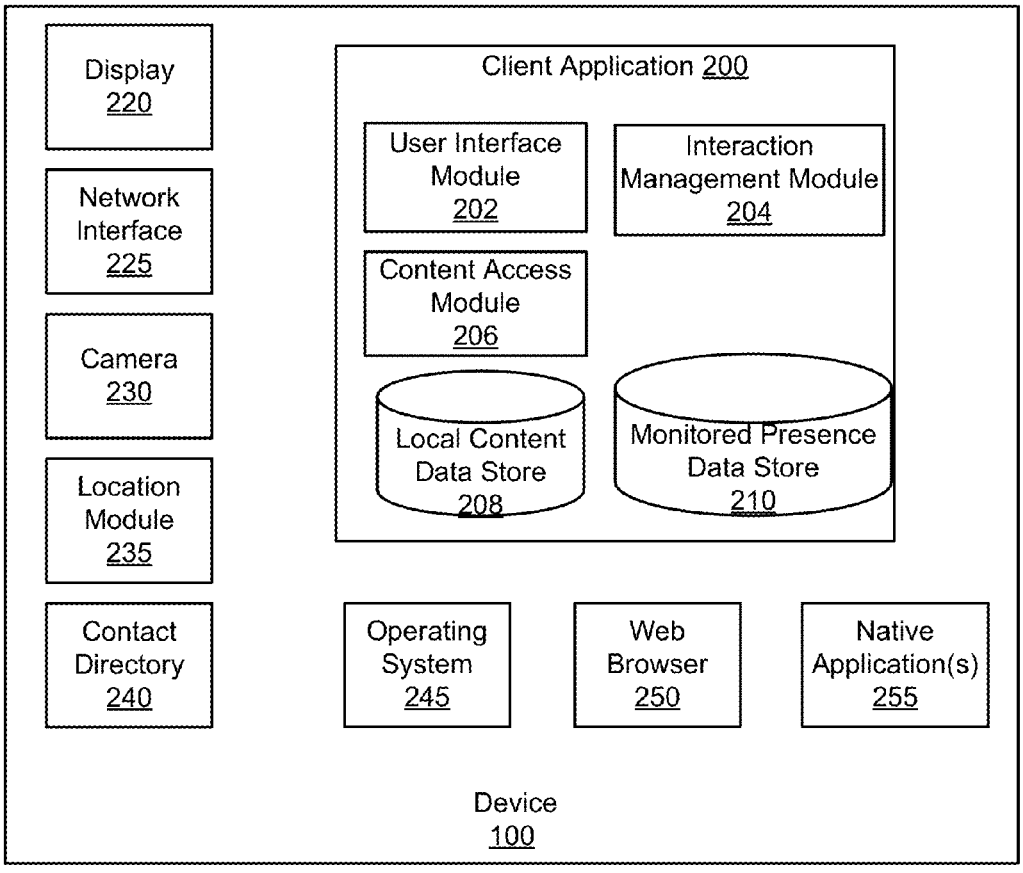

Display
220
Network
Interface
225
Camera
230
Location
Module
235
Contact
Directory
240
Client Application 200
User Interface
Module
202
Interaction
Management Module
204
Content Access
Module
206
Local Content
Data Store
208
Monitored Presence
Data Store
210
Operating
System
245
Web
Browser
250
Native
Application(s)
255
Device
100

Fig. 3A

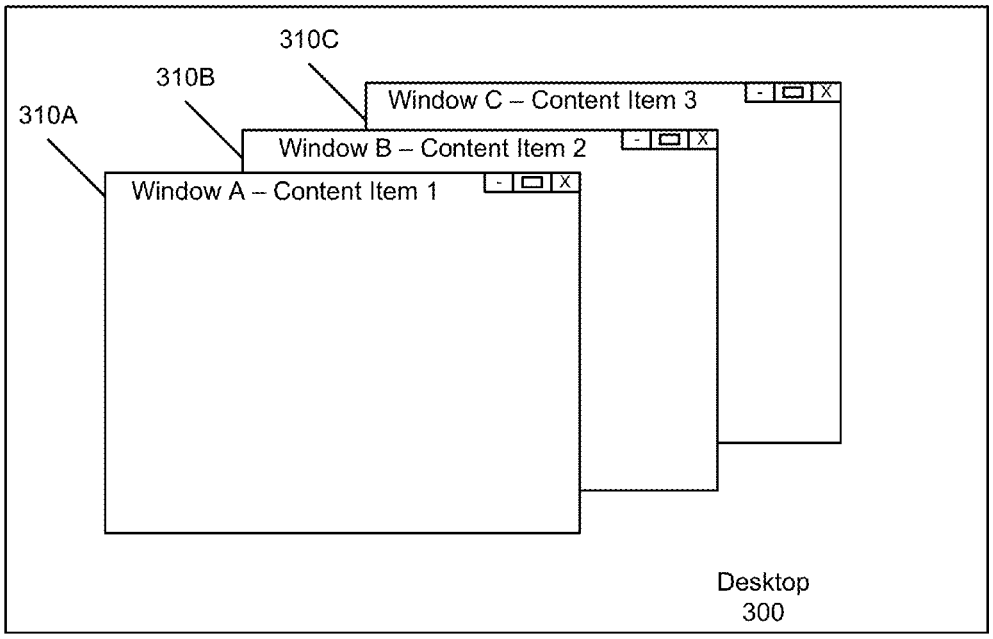
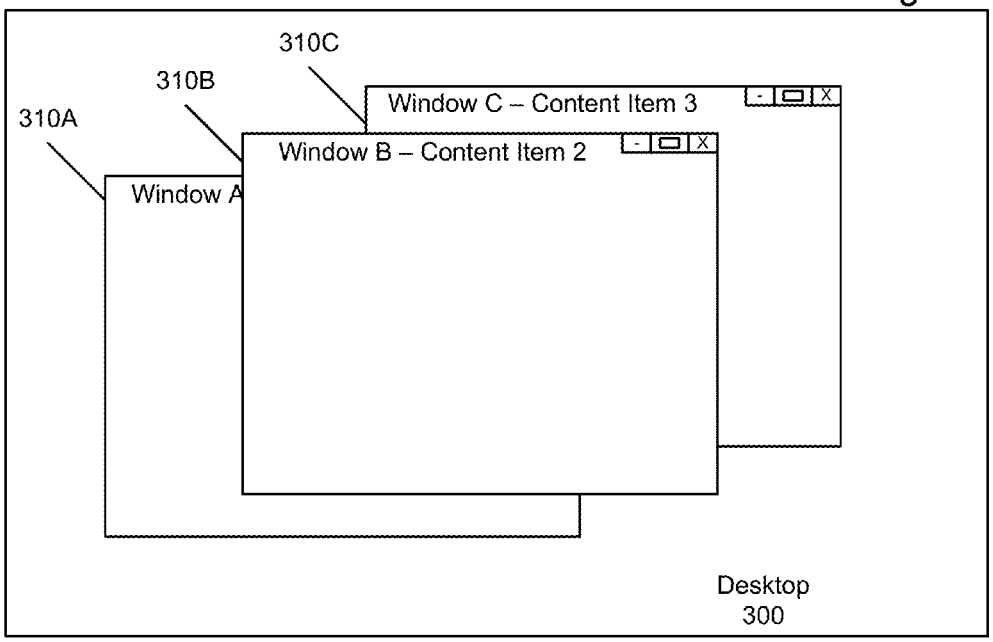

310C
310B
310A
Window C – Content Item 3
Window B – Content Item 2
Window A – Content Item 1
Desktop
300

Fig. 3B


310C
310B
310A
Window C – Content Item 3
Window B – Content Item 2
Window A
Desktop
300

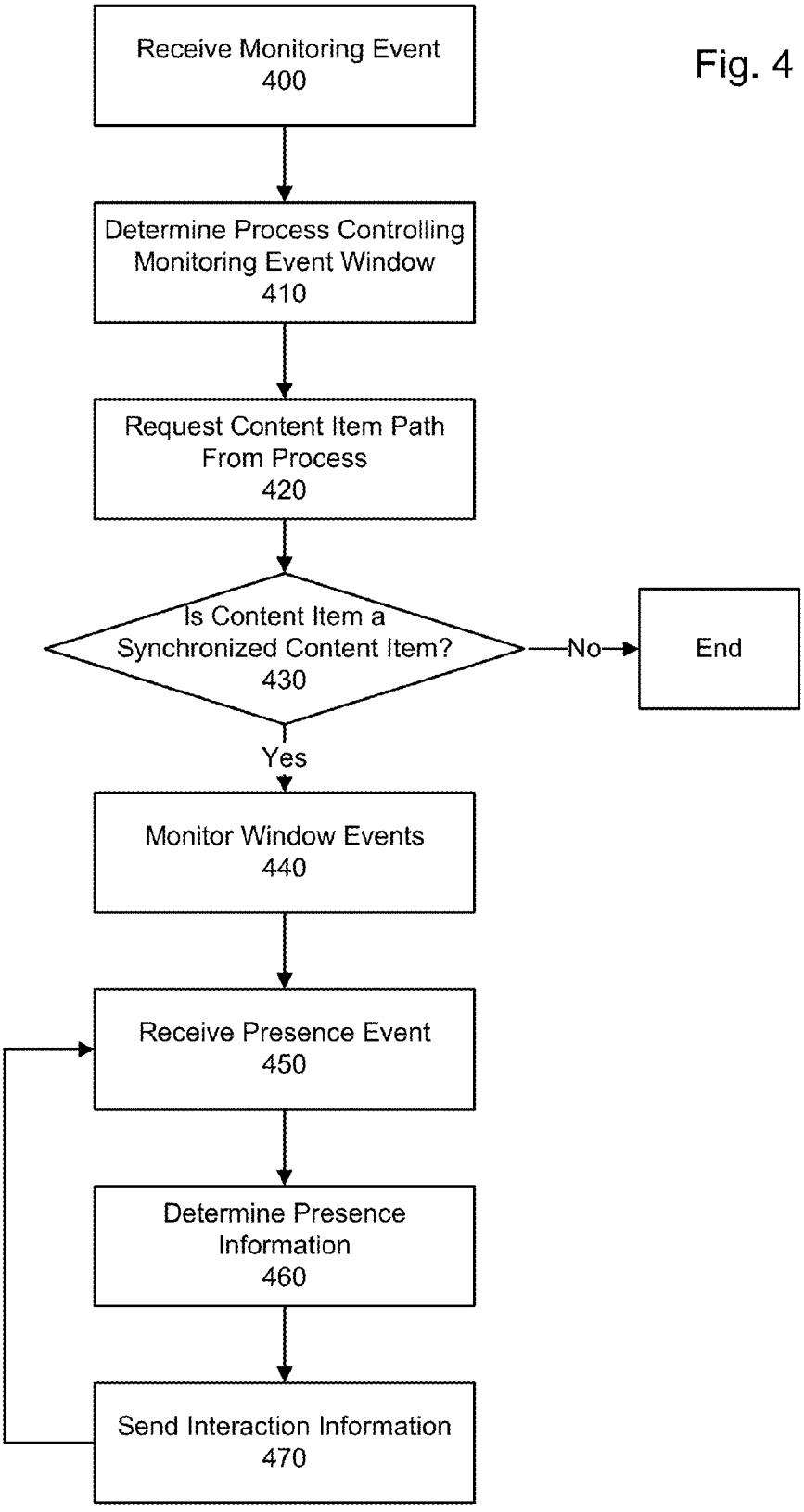

Receive Monitoring Event
400
Determine Process Controlling Monitoring Event Window
410
Request Content Item Path From Process
420
Is Content Item a Synchronized Content Item?
430
No
End
Yes
Monitor Window Events
440
Receive Presence Event
450
Determine Presence Information
460
Send Interaction Information
470

Fig. 4

Fig. 5A

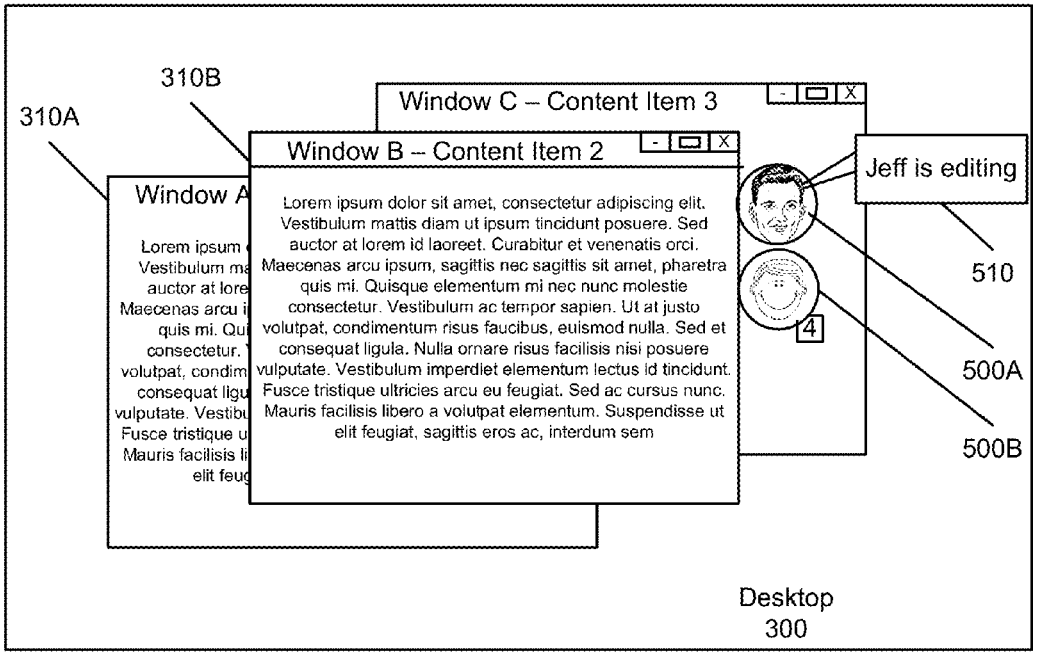

310B
310A
Window C – Content Item 3
Window B – Content Item 2
Window A
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum mattis diam ut ipsum tincidunt posuere. Sed auctor at lorem id laoreet. Curabitur et venenatis orci. Maecenas arcu ipsum, sagittis nec sagittis sit amet, pharetra quis mi. Quisque elementum mi nec nunc molestie consectetur. Vestibulum ac tempor sapien. Ut at justo volutpat, condimentum risus faucibus, euismod nulla. Sed et consequat ligula. Nulla ornare risus facilisis nisi posuere vulputate. Vestibulum imperdiet elementum lectus id tincidunt. Fusce tristique ultricies arcu eu feugiat. Sed ac cursus nunc. Mauris facilisis libero a volutpat elementum. Suspendisse ut elit feugiat, sagittis eros ac, interdum sem
Jeff is editing
510
4
500A
500B
Desktop
300

Fig. 5B

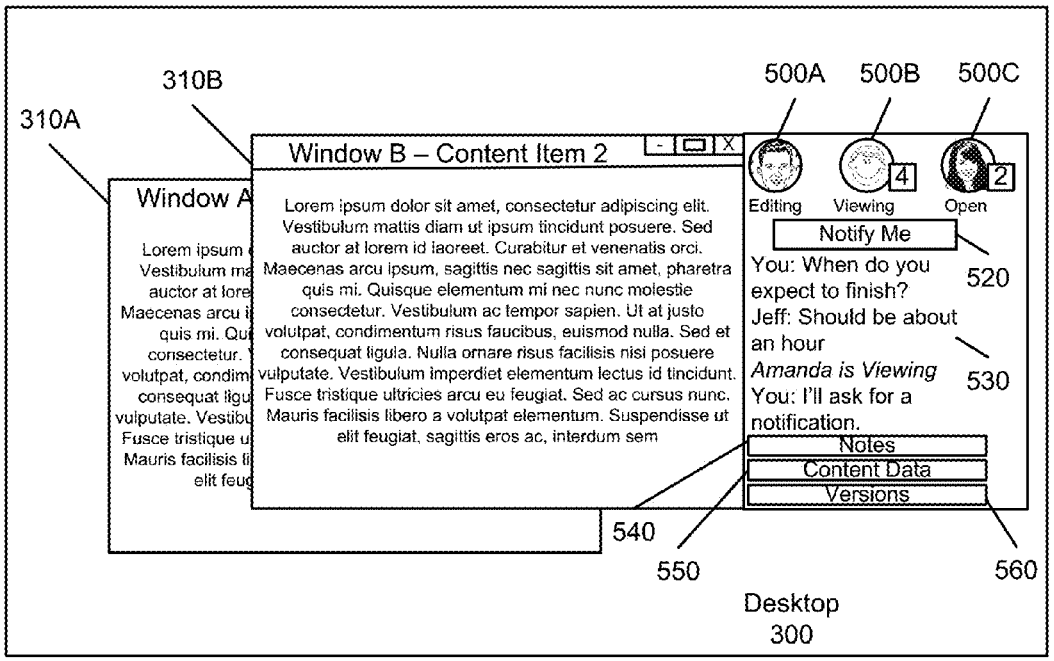

310B
310A
500A
500B
500C
Window B – Content Item 2
Window A
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum mattis diam ut ipsum tincidunt posuere. Sed auctor at lorem id laoreet. Curabitur et venenatis orci. Maecenas arcu ipsum, sagittis nec sagittis sit amet, pharetra quis mi. Quisque elementum mi nec nunc molestie consectetur. Vestibulum ac tempor sapien. Ut at justo volutpat, condimentum risus faucibus, euismod nulla. Sed et consequat ligula. Nulla ornare risus facilisis nisi posuere vulputate. Vestibulum imperdiet elementum lectus id tincidunt. Fusce tristique ultricies arcu eu feugiat. Sed ac cursus nunc. Mauris facilisis libero a volutpat elementum. Suspendisse ut elit feugiat, sagittis eros ac, interdum sem
Editing
Viewing
4
Open
2
Notify Me
520
You: When do you expect to finish?
Jeff: Should be about an hour
Amanda is Viewing
530
You: I'll ask for a notification.
Notes
Content Data
Versions
540
550
560
Desktop
300

Fig. 5C

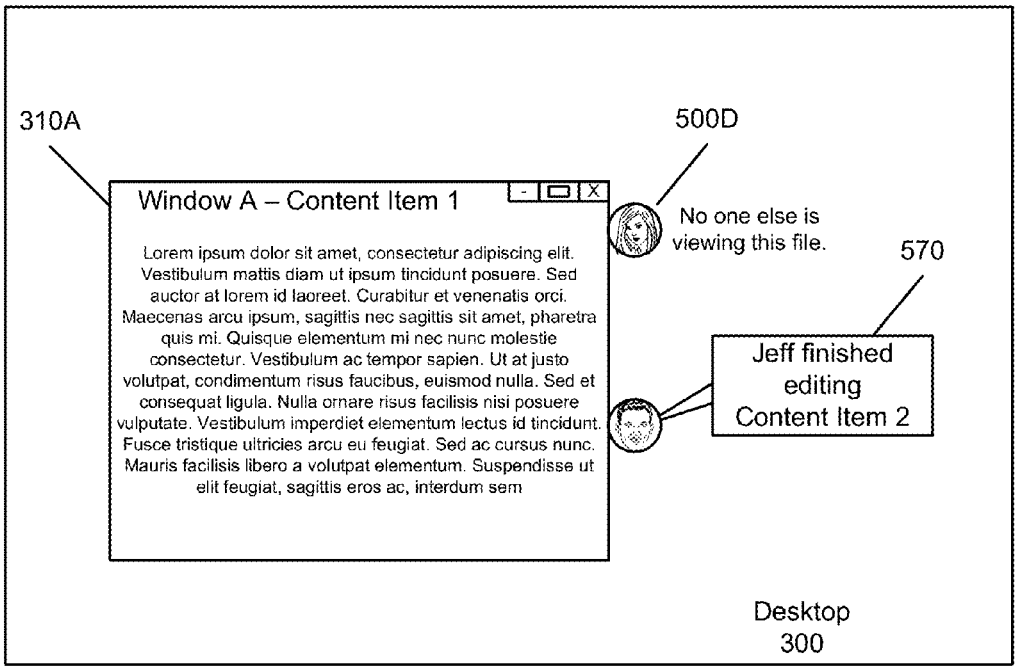

310A
500D
Window A – Content Item 1
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum mattis diam ut ipsum tincidunt posuere. Sed auctor at lorem id laoreet. Curabitur et venenatis orci. Maecenas arcu ipsum, sagittis nec sagittis sit amet, pharetra quis mi. Quisque elementum mi nec nunc molestie consectetur. Vestibulum ac tempor sapien. Ut at justo volutpat, condimentum risus faucibus, euismod nulla. Sed et consequat ligula. Nulla ornare risus facilisis nisi posuere vulputate. Vestibulum imperdiet elementum lectus id tincidunt. Fusce tristique ultricies arcu eu feugiat. Sed ac cursus nunc. Mauris facilisis libero a volutpat elementum. Suspendisse ut elit feugiat, sagittis eros ac, interdum sem
No one else is viewing this file.
570
Jeff finished editing Content Item 2
Desktop 300

Fig. 5D

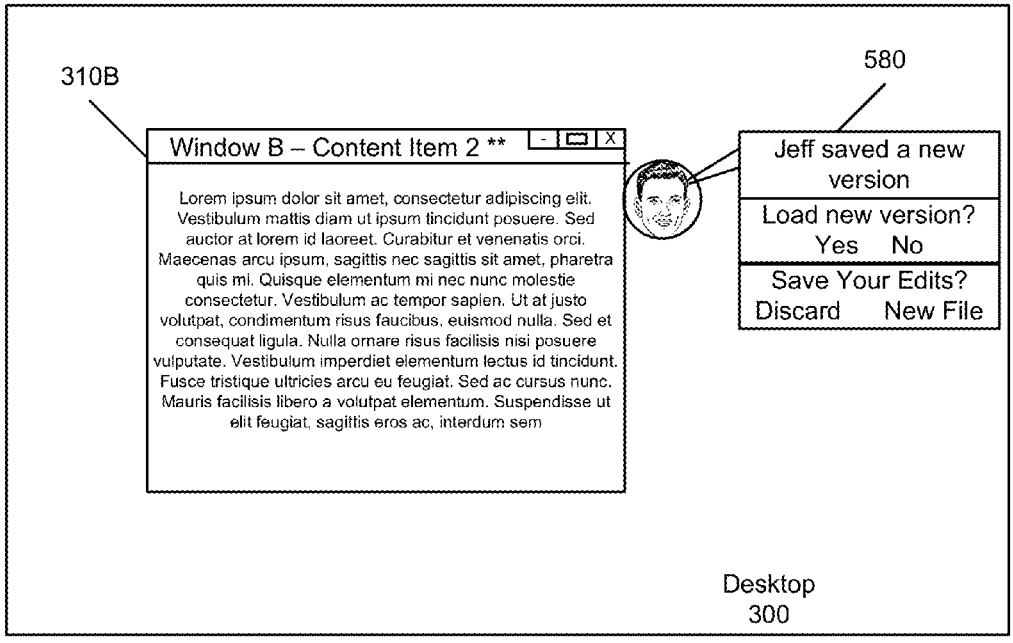

310B
580
Window B – Content Item 2 **
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum mattis diam ut ipsum tincidunt posuere. Sed auctor at lorem id laoreet. Curabitur et venenatis orci. Maecenas arcu ipsum, sagittis nec sagittis sit amet, pharetra quis mi. Quisque elementum mi nec nunc molestie consectetur. Vestibulum ac tempor sapien. Ut at justo volutpat, condimentum risus faucibus, euismod nulla. Sed et consequat ligula. Nulla ornare risus facilisis nisi posuere vulputate. Vestibulum imperdiet elementum lectus id tincidunt. Fusce tristique ultricies arcu eu feugiat. Sed ac cursus nunc. Mauris facilisis libero a volutpat elementum. Suspendisse ut elit feugiat, sagittis eros ac, interdum sem
Jeff saved a new version
Load new version?
Yes No
Save Your Edits?
Discard New File
Desktop
300

Fig. 6

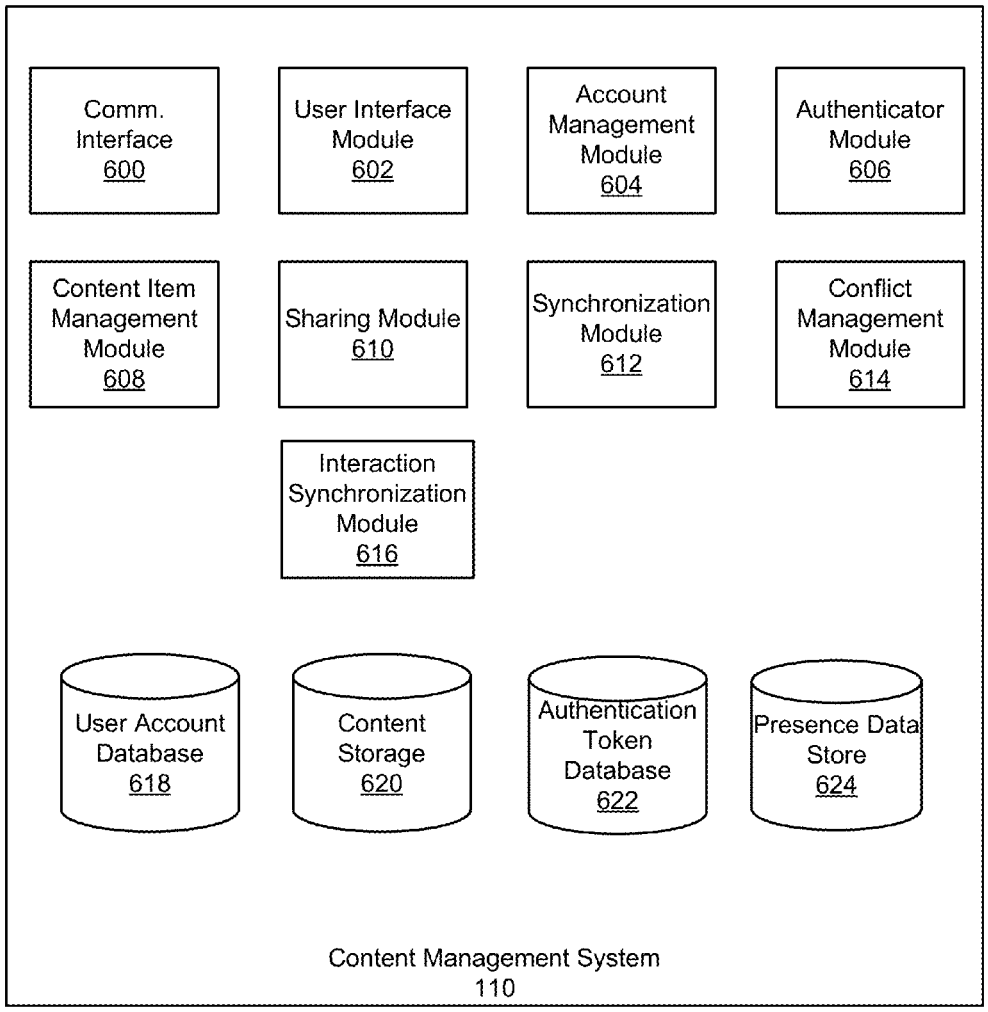

Comm. Interface 600
User Interface Module 602
Account Management Module 604
Authenticator Module 606
Content Item Management Module 608
Sharing Module 610
Synchronization Module 612
Conflict Management Module 614
Interaction Synchronization Module 616
User Account Database 618
Content Storage 620
Authentication Token Database 622
Presence Data Store 624
Content Management System 110

Fig. 7

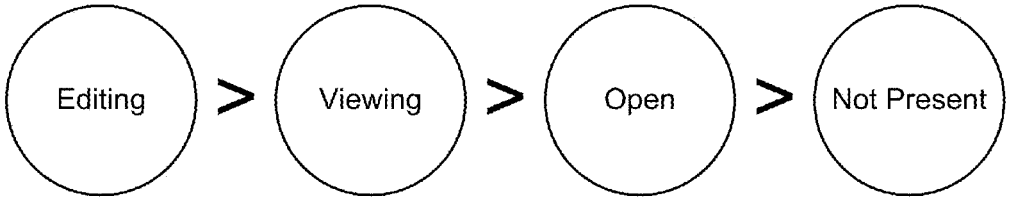


Fig. 8

800

Content Item 27cd0162a6144bf				
Device ID	UserID	User Interface Element ID	Process ID	Presence Information
47a02b4	Jim	5	87	View
2581a26	Jim	7	128	Open
36abe87	John	15	23	Edit
36abe87	John	10	23	Open
9d7523c	John	1	7	View
68792aa	Sue	92	1078	Open
47a02b4	Amanda			Notify (John ends editing)

810

UserID	Presence
Jim	View
John	Edit
Sue	Open

Fig. 9

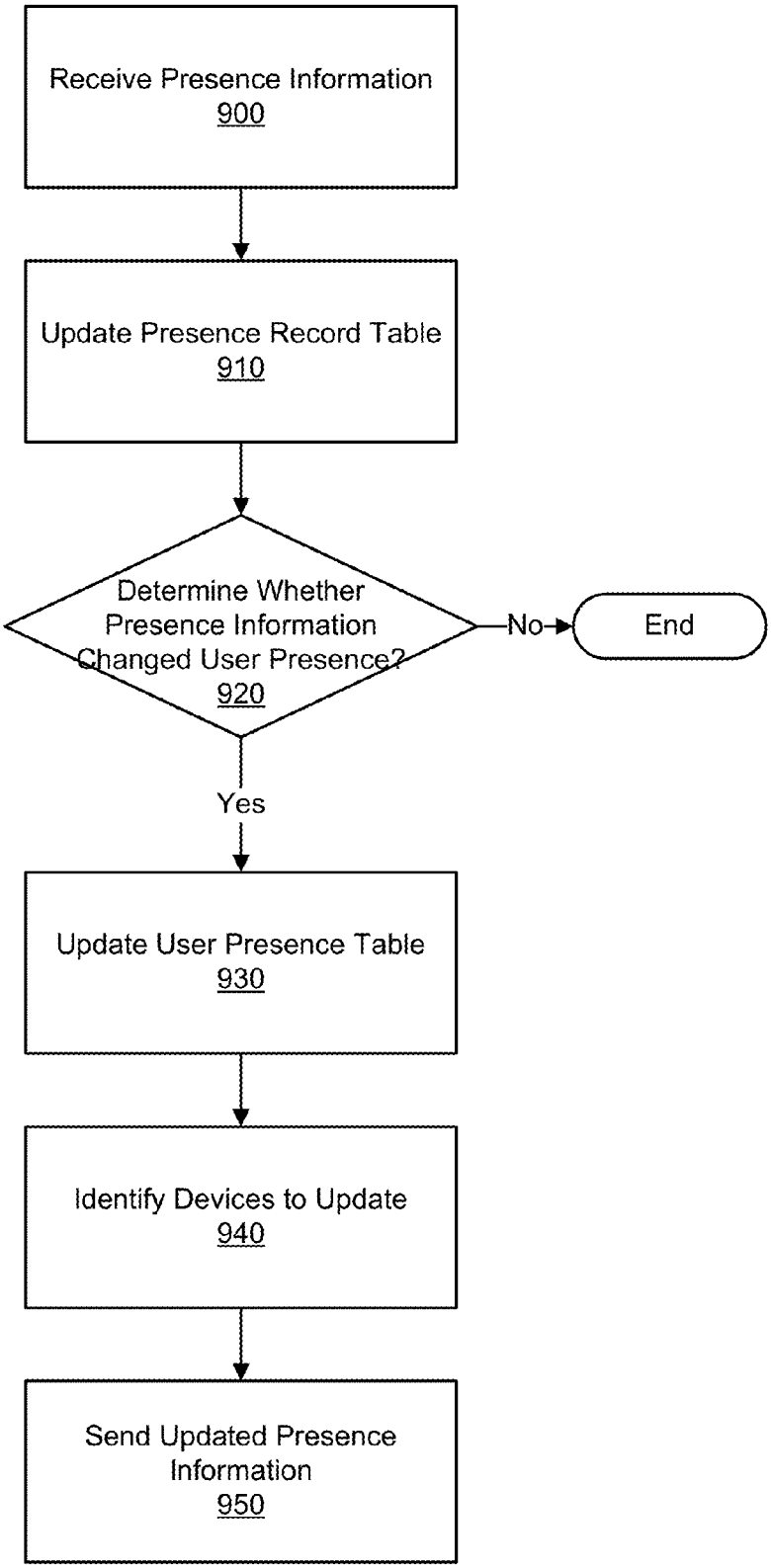

Receive Presence Information
900
Update Presence Record Table
910
Determine Whether Presence Information Changed User Presence?
920
No
End
Yes
Update User Presence Table
930
Identify Devices to Update
940
Send Updated Presence Information
950

Fig. 10

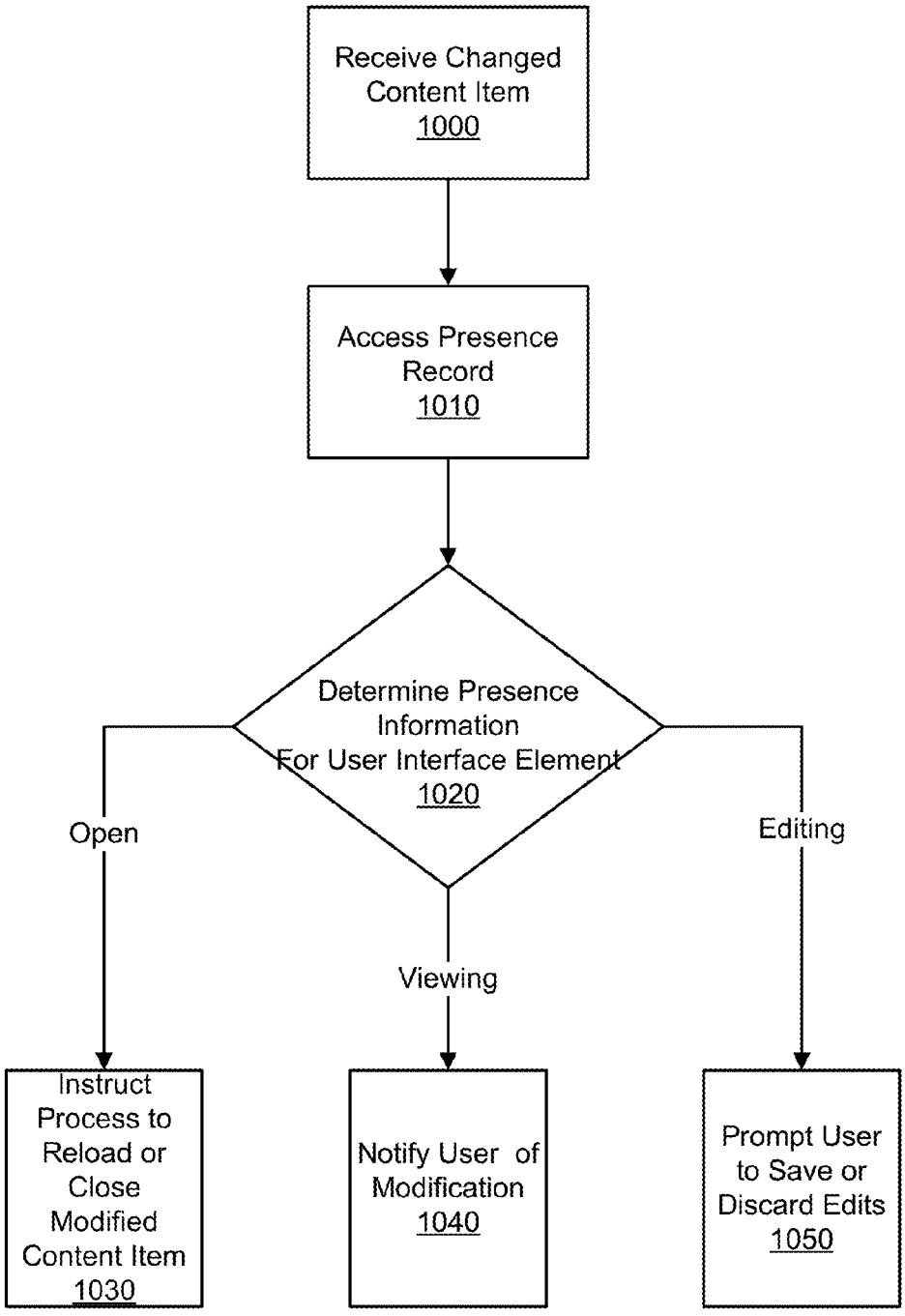
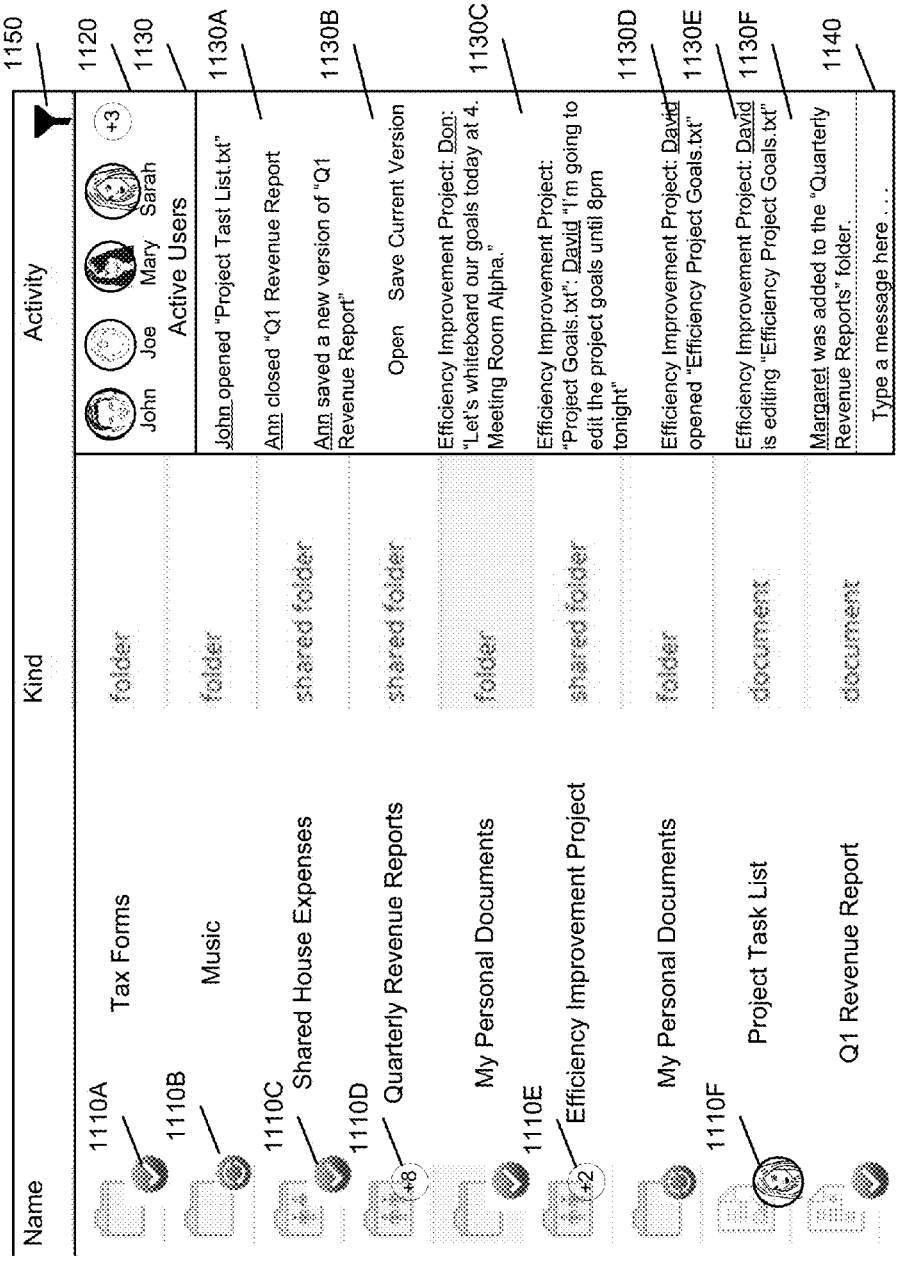

Receive Changed Content Item 1000
Access Presence Record 1010
Determine Presence Information For User Interface Element 1020
Open
Viewing
Editing
Instruct Process to Reload or Close Modified Content Item 1030
Notify User of Modification 1040
Prompt User to Save or Discard Edits 1050

Fig. 11


1100
My Workspace
Name
Kind
Activity
1150
1120
1130
1110A
Tax Forms
folder
1110B
Music
folder
1110C
Shared House Expenses
shared folder
1110D
Quarterly Revenue Reports
shared folder
My Personal Documents
folder
1110E
Efficiency Improvement Project
shared folder
My Personal Documents
folder
1110F
Project Task List
document
Q1 Revenue Report
document
John
Joe
Mary
Sarah
+3
Active Users
John opened "Project Tast List.txt"
1130A
Ann closed "Q1 Revenue Report
Ann saved a new version of "Q1 Revenue Report"
1130B
Open Save Current Version
Efficiency Improvement Project: Don: "Let's whiteboard our goals today at 4. Meeting Room Alpha."
1130C
Efficiency Improvement Project: "Project Goals.txt": David "I'm going to edit the project goals until 8pm tonight"
1130D
Efficiency Improvement Project: David opened "Efficiency Project Goals.txt"
1130E
Efficiency Improvement Project: David is editing "Efficiency Project Goals.txt"
1130F
Margaret was added to the "Quarterly Revenue Reports" folder.
1140
Type a message here . . .

DETERMINING PRESENCE IN AN APPLICATION ACCESSING SHARED AND SYNCHRONIZED CONTENT

BACKGROUND

This disclosure relates generally to sharing information among devices, and particularly to sharing information between devices about native application interactions involving shared and synchronized content items.

Content management systems permit devices to synchronize content items with the content management system and other devices. A device stores a local copy of content items. When content items are added, deleted, and edited on a device, these modifications are sent to the content management system for storage and synchronization with other devices. To interact with a content item, users typically execute a native application on the device to view and modify the content item. Modifications to a content item may be synchronized with the content management system separately from the execution of the native application. Accordingly, multiple devices may separately view and edit a particular content item. When users each modify the same content item, versioning problems may arise from conflicting edits. In part, these conflicts arise because users are not aware that other users are modifying the contact item in parallel.

SUMMARY

Described embodiments enable a set of users of devices sharing content items via a content management system to be aware of each other's interactions with those shared items on their respective devices. In various embodiments graphical and textual information is provided to users on their respective devices indicating that a shared content item is being viewed or edited on one or more other devices. In addition, in various embodiments users can communicate their comments via their respective devices and register to receive notifications with respect to the shared content items.

A device in accordance with various embodiments stores a local copy of a shared content item, which is maintained and synchronized between devices by a content management system. The device includes a native application that can be used to access the content item, such as a word processor, media viewer, media editor, and so forth. The native application displays information relating to the content item in a user interface element, such as a window. The device also includes a client application that monitors interactions with the content item and communicates information about those interactions to other devices sharing the content item either directly or via the content management system. The client application generates interaction information including data describing the user's interactions with a content item. Interaction information includes interactions with the client application and interactions with the native application. Interaction information may be determined from programmatic events that occur in the native application. Interaction information determined from events of the native application is termed presence information. Events occurring in the native application interactions are termed presence events and can include opening a content item, editing a content item, saving a content item, renaming a content item, moving a content item, and deleting a content item. Presence events also include interactions with the user interface element of the native application, such as the user interface element gaining or losing focus. A focused user interface element is the user interface element that receives user inputs at the device. For example, a user interface element having focus is used to determine presence information that a user is "viewing" the content item via that user interface element. In one embodiment, the presence events are gathered by the client application that is separate from the native application interacting with the content item. That is, in this embodiment the presence events are gathered by another application or process that is not integrated into the native application accessing the content item.

Additional types of interaction information include notes, messages, and notification requests related to the content item, which may be received by the client application. Messages may include chat messages to other devices, and messages indicating a user's intent to interact with (e.g., to edit) a content item. Interaction information also includes metadata modifications, such as versioning notes, or requests for further information stored at the content management system about the content item, such as a request to view versioning information or prior content item versions. The interaction information is exchanged by the devices to provide users with information regarding others' interactions with the content item.

The client application monitors events on the device to determine when a user is using a native application to interact with a synchronized content item. In one method of monitoring events, the client application registers with a process controlling a user interface element or the operating system to receive events related to the user interface element. The client application may monitor all such events, or only certain user interface elements associated with opening a synchronized content item. When a presence event occurs, the client application receives the presence event from the native application or from an operating system. The client application identifies a user interface element associated with the event and determines presence information associated with the event. The client application notifies the content management system that the content item is being or has been interacted with, and sends the presence information to the content management system. The presence information (or other interaction information) may be sent directly to other devices.

The content management system receives the interaction information and may respond to the interaction information or send the interaction information to other devices based on the type of interaction information. Interaction information relating to messages is sent to other devices synchronized to the content item. Presence information is stored as a presence record, which in one embodiment indicates a content item, device, process, user interface element, and a type of presence (e.g., that the content item is open, viewed, or being edited). A user presence may be determined that indicates a priority of interaction according to an ordering of interactions. In an example ordering, editing a content item has a higher priority than viewing the content item, which has a higher priority than opening the content item. The user presence describes a user's presence with respect to a content item and may be without reference to any particular device, process, or user interface element. The content management system may send the information to all devices that are synchronized with respect to the content item, or just to devices that are associated with an active presence record with respect to the content item. In one embodiment, devices register with the content management system to receive interaction information about a content item, which may specify a particular interaction to trigger notification. When

the interaction occurs, the triggering interaction information is sent to devices associated with the notification.

In one embodiment, when modifications to a content item are received at the content management system, devices are instructed to perform actions depending on the interaction information associated with the client device. For example, a device associated with a presence record is notified that the content item is modified, and the content management system instructs the device to display a prompt for the user to select whether to discard the content item edits or create a new version of the content item.

When presence information for other client devices is received that relates to a content item opened by a native application, the client application displays the presence information in a presence indicator near the user interface element of the native application. The presence indicator may be an image of a user associated with the received presence information, and may indicate the particular type of the presence information, and the number of users exhibiting that type of presence. To display the presence indicator, the client device determines the boundaries of the user interface element of the native application and displays the presence indicator at the boundary of the user interface element or in another suitable location to not obscure the user interface element. For example, the presence indicator may be a thumbnail image of the user modifying a shared content item, displayed alongside a border of the window containing the content item. The presence indicator may also be displayed on a toolbar or other portion of a display. In addition to the presence information, other types of interaction information may also be presented. For example, chat messages, a requested notification about a user's presence, versioning information, and other interaction information may be displayed. When a new version of a content item is synchronized to the content management system, the interface may also display information informing the user to update the content item and providing options based on the presence information of the user interface element, for example by providing the user an option to save or discard changes to a prior version of the content item.

Devices display presence information relating to other users' interactions with a content item along with a user interface of the application displaying the content item. This permits users to view, in essentially real or near real-time, the activity of other users with respect to a content item. In addition, the presence information is collected in one embodiment without relying on modifications or add-ons to the native application displaying or modifying the content item. In addition, presenting interfaces to communicate with the content management system alongside the application user interface permits users to conveniently exchange chat information and retrieve data about a content item while the content item is being displayed.

In various embodiments, the client application additionally displays presence information in a content item browser ("browser") for viewing a folder or other collection of content items. The collection of content items may include content items that may be interacted with by a native application, and may include collections within the collection. Thus, each collection may itself be a collection of content items, such that the collection of content items may be hierarchically arranged (e.g., a folder within a folder). Interaction information for a particular content item is associated with each collection of content items that contain the content item.

For clarity of description, when the browser displays a collection that includes a collection (i.e., the folder within a folder), the included collection is termed an organizational element. That is, when the browser displays a collection, the displayed content items that are collections are termed organizational elements.

When a user views a collection of content items in the browser, the browser displays a status indicator reflecting interaction information related to the content items. Each organizational element displayed in the browser also displays an indicator reflecting interaction information of content items, such as further organizational elements, within that organizational elements. For example, a browser may display a collection of two content items and one organizational element (e.g. a folder containing three content items). A status indicator is displayed with each of the two content items to indicate interaction information relating to the content items, and a status indicator is displayed with the organizational element indicating interactions with its content items (e.g., the three content items within the folder). To obtain further information, a user may select or place a cursor over a content item or organizational element. Alternatively, the user may control the browser to display the collection of content items described by an organizational element. To generate the display of the browser, the client application filters the interaction information to display interaction information relevant to the organization of content items being displayed to the user. This permits a user to view, within a content item browser, interaction information for the content items associated with the organization of content items, and to view, at a glance, interactions of other users with respect to various organizational elements and content items.

Interaction information and other metadata relating to the collection may also be displayed to the viewing user in an activity feed. The interaction information and other metadata may be associated with a shared folder that is synchronized with the client and includes the displayed collection. The activity feed displays interaction information and other information associated with the collection of content items presently displayed in the browser. The activity feed may display currently active users (e.g., those with active presence information, such as "viewing"), along with actions performed by those users and notifications related to the collection of content items. These actions and notifications may include changes in presence information, or other notifications relating to interaction information, such as the interaction information described above. In particular, this may include interaction information entered by a user using an interface element displayed with a native application, such as a chat message or a request to be notified. In addition to the interaction information, the activity feed may also display information reflecting changes to content items, the collection, or (if applicable) the shared folder. For example, the changes to the shared folder may be a new user added to the shared folder or a change to a content item may be a new version of a content item becoming available. The user may also interact with the browser to filter for particular users (e.g., to identify a content item, including an organization element, where a user is associated with presence information), to identify when a user last interacted with a content item, or to add interaction information (e.g., a chat message) relating to the organizing element itself.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows an embodiment of an environment for content item synchronization including communication of interaction information.

FIG. **2** shows various modules and components of a device in accordance with one embodiment.

FIGS. **3**A and **3**B show a user interface element focus change on a desktop display of a device.

FIG. **4** shows an example process for determining presence information associated with a content item according to one embodiment.

FIGS. **5**A-**5**D show example user interfaces displaying interaction information.

FIG. **6** shows components of a content management system, according to one embodiment.

FIG. **7** shows an ordering of user presence according to one embodiment.

FIG. **8** shows a presence record in a presence table relating to a content item according to one embodiment.

FIG. **9** shows a process of modifying a presence record and updating devices of user presence according to an embodiment.

FIG. **10** shows a flowchart for a process that uses presence information to synchronize modifications to a content item.

FIG. **11** shows a folder-level display of interaction information according to one embodiment.

The figures depict various embodiments of the present invention for purposes of illustration only. One skilled in the art will readily recognize from the following discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention described herein.

DETAILED DESCRIPTION

FIG. **1** shows an embodiment of an environment for content item synchronization including communication of interaction information. FIG. **1** includes devices **100**A, **100**B, **100**C (referred to generally as device **100**), content management system **110**, and network **120**. Three devices are shown only for purpose of illustration; in practice any number of devices may be present in the environment. Similarly, other modules or components described and illustrated throughout may include single or multiple instances as appropriate to the needs of the implementer and without loss of generality.

Device **100** may be any suitable computing device for locally storing and viewing content items and synchronizing the content items with content management system **110**. Examples of devices include desktop and laptop computers, hand-held mobile devices, tablet computers, and other computing devices. The operation of device **100** in various embodiments is further described below.

Each device **100** communicates with content management system **110** through network **120**. Network **120** is any suitable network and may include local networks, corporate networks, global networks, and any combination of these. In typical configurations, devices **100** communicate via a wired or wireless communication network to a local network service provider, and communicate with content management system **110** through the Internet. In certain configurations, devices **100**A, **100**B, and **100**C communicate directly with one another without network **120** as indicated in FIG. **1** by dashed lines. For example, devices **100** may communicate via a wired or wireless connection, such as wirelessly via a Bluetooth connection or a wired connection via a Universal Serial Bus (USB).

Content management system **110** provides content sharing and synchronization services for users of devices **100**. These services allow users to share content with users of other devices **100**. In addition to content sharing, content management system **110** updates shared content responsive to changes and enables synchronized changes to content items across multiple devices **100**. A user may synchronize content across multiple devices **100** owned by the user and associated with the user's account, and the user may share content that is synchronized with devices associated with other users' accounts. Content stored by content management system **110** can include any type of data, such as digital data, documents, media (e.g., images, photos, videos, audio, streaming content), data files and databases, source and object code, recordings, and any other type of data or file, collectively referred to here as "content items." Content items stored by content management system **110** may also be used to organize other content items, such as folders, tables, collections, albums, playlists, or in other database structures (e.g., object oriented, key/value etc.). In practice, various devices **100** may be synchronizing different groups of content items, based on user associations, permissions, content sharing permissions, and so forth. The operation of content management system **110** in various embodiments is further described below.

FIG. **2** shows various modules and components of device **100** in accordance with one embodiment. Device **100** includes display **220** for providing information to the user, and in certain client devices **100** includes a touchscreen. Device **100** also includes network interface **225** for communicating with content management system **110** via network **120**. Other conventional components of a client device **100** that are not material are not shown, for example one or more computer processors, local fixed memory (RAM and ROM), as well as optionally removable memory (e.g., SD-card), power sources, and audio-video outputs.

Software modules include operating system **245** and one or more native applications **255**. Native applications **255** vary based on the client device, and may include various applications for creating, viewing, consuming, and modifying content stored on content management system **110**, such as word processors, spreadsheets, database management systems, code editors, image and video editors, e-book readers, audio and video players, and the like. Operating system **245** on each device provides a local file management system and executes the various software modules such as content management system client application **200** and native application **255**. A contact directory **240** stores information about the user's contacts, such as name, picture, telephone numbers, company, email addresses, physical address, website URLs, and the like. Further operation of native applications **255**, operating system **245**, and content management system client application **200** are described below.

In certain embodiments, device **100** includes additional components such as camera **230** and location module **235**. Camera **230** may be used to capture images or video for upload to the online content management system **110**. Location module **235** determines the location of device **100**, using for example a global positioning satellite signal, cellular tower triangulation, or other methods. Location module **235** may be used by client application **200** to obtain location data and add the location data to metadata about a content item, such as an image captured by camera **230**.

Client device **100** accesses content management system **110** in a variety of ways. Client application **200** can be a dedicated application or module that provides access to the services of content management system **110**, providing both user access to shared files through a user interface, as well as programmatic access for other applications. Client device **100** may also access content management system **110**

through web browser **250**. As an alternative, client application **200** may integrate access to content management system **110** with the local file management system provided by operating system **245**. When access to content management system **110** is integrated in the local file management system, a file organization scheme maintained at content management system **110** is represented as a local file structure by operating system **245** in conjunction with client application **200**. Client application **200** may take various forms, such as a stand-alone application, an application plug-in, or a browser extension. Client application **200** includes user interface module **202**, interaction management module **204**, content access module **206**, local content data store **208**, and monitored presence data store **210**.

In addition to handling other device tasks, operating system **245** displays information from applications executing on device **100** to a user via display **220**, which may include one or more user interface elements. Such user interface elements may vary based on the particular device and configuration. User interface elements include windows on a desktop interface as well as interface elements on a mobile device. Examples of operating systems that employ user interface elements such as windows are Microsoft Windows 8 by Microsoft Corporation of Redmond, Wash., and OS X by Apple Inc. of Cupertino, Calif. In addition, operating system **245** manages control of multiple native applications **255**, which may be executing simultaneously. The user interface elements may be layered, such that one layer overlaps another layer. In some operating systems and configurations, only a single user interface element is displayed at a given time. One user interface element is typically the active user interface element, meaning that it is the user interface element to which the operating system **245** routes user inputs, such as keyboard entry, cursor movement, touch sensors, touch gestures, and so forth. As understood by those of skill in the art, a window or other user interface element that is active at a particular time is often said to have focus. Users may select another user interface element to change the focus from one user interface element to another, and in some instances operating system **245** may change the focus without user input.

Typically, the user interface elements, e.g., windows, associated with native applications **255** are managed by operating system **245**, which maintains an association between process identifiers of executing native applications **255** and user interface element identifiers of the user interface elements. For example, a particular application may be associated with process id "2587", which may be managing multiple user interface elements, with user interface element identifiers **4**, **8**, and **10**. Each user interface element identifier may be separately associated with a particular content item opened by that native application **255**, and multiple user interface element identifiers and process identifiers may be associated with the same content item.

Operating system **245** also handles and recognizes various events. Such events include a request from native applications **255** to close or open a content item, a request from native applications **255** to close a window or other user interface element, and requests to change a user interface element focus, among many others. As described below, these events may be used by interaction management module **204** to recognize a change in presence related to a content item.

Client application **200** identifies interactions that take place with respect to a content item, such as when a user opens, closes or edits the content item on the device. These interactions are identified by client application **200** to generate interaction information describing the interaction with the content item. Interaction information includes interactions with client application **200** and interactions with native application **255**. Interaction information determined from actions of native application **255** is termed presence information. An application, such as client application **200** that determines interaction information and presence information is termed a presence application. Additional types of interaction information (in addition to presence information) include notes, messages, and notification requests related to the content item, which may be received by client application **200**. Messages may include chat messages to other devices, and messages indicating a user's intent to interact with (e.g., to edit) a content item. Notification requests may include a request to be notified when another user's interaction information changes. Interaction information also includes metadata modifications, such as versioning notes, or requests for further information stored at content management system **110** about the content item, such as a request to view versioning information or prior content item versions. Further examples of interaction information is described below.

This interaction information is transmitted to other devices **100** that are synchronized with respect to the content item. The indication of intent may for example alert a second user of the content item on another device that the first user would like to edit the content item. Client application **200** identifies when users are using a native application **255** to interact with a content item, and also receives chat or intent information from a user. In various embodiments, device **100** identifies a user's presence in a content item (i.e. that the user has the content item open or is editing the content item) through interaction with operating system **245** as described further below.

Device **100** receives content items from content management system **110** and permits users to view, modify, and interact with the content items using various native applications **255** stored on the device **100**. For example, device **100** may include a photo editing application that manipulates image content items, or a word processing application that permits modification of text content items. As described further below, interaction information is determined by device **100** using user interactions applications and the interaction information is sent to other devices **100**. In addition, when device **100** receives interaction information relating to other devices **100**, the device **100** displays that interaction information.

In one embodiment, an application detecting interaction information relating to content items is distinct from the applications viewing or manipulating the content items. For example, the client application detecting interaction information is distinct from a photo editing application manipulating or displaying the image content items. In various embodiments, the application detecting interaction information is also responsible for synchronizing the content items with content management system **110**. Since the application detecting presence information may be distinct from the applications about which presence is detected, presence may be monitored for many applications and content items at once and without requiring integration of the presence monitoring into each type of content item viewer. That is, no special presence monitoring add-on or application modification is required, for example, for each of a photo editing application, a word processing application, and a playlist editing application.

FIGS. **3**A and **3**B show an example of a user interface element focus change on desktop **300** shown on display **220**

of device **100**. In FIG. **3**A, windows **310**A, **310**B, and **310**C are displayed on desktop **300** and viewable by the user. In this embodiment, desktop **300** is a general container or frame maintained by operating system **245** that encloses user interface elements on display **220**. In FIGS. **3**A and **3**B, the user interface elements are windows **310** in a desktop computing environment. In other configurations, such as a mobile device, or other display with limited area, only a single user interface element might be displayed at a time. As shown by FIG. **3**A, window **310**A is the active window, shown as the front window, partially obscuring windows **310**B and **310**C. In FIG. **3**B, focus changed to window **310**B, which is now the front window and the active window. The focus may change due to user interaction with window **310**B, or due to a process requesting that its window become the active window. In certain operating systems and configurations, a user interface element has focus (e.g., is receiving user input) without being the front user interface element.

Referring again to FIG. **2**, to open a content item, native application **255** requests the content item from operating system **245** and receives a handle to the content item from operating system **245** for the content item. In some cases, application **245** does not maintain the handle, and may load the content item data into memory and subsequently close the content item handle even if native application **255** continues to use data from the content item or if the user enters edits to the content item. Accordingly, open content item handles are often not a reliable way to determine whether an application is interacting with a particular content item. As such, in certain embodiments, further behaviors exhibited by the native applications **255** are used to determine whether an application is editing a content item.

Native applications **255** also perform various behaviors when a user modifies a content item, and prior to the user saving the content item. These behaviors vary based on the application and operating system **245**. For example, some native applications **255** create a temporary content item with a filename that differs from the open content item, for example leading the temporary content item's filename with a tilde or other recognizable mark. In other examples, the native applications **255** changes the title of a user interface element associated with the content item, which may or may not be directly viewable by a user. In still further examples, native application **255** sets a flag indicating the content item has been modified. Native application **255** may also provide information regarding content item modification in response to a request from another application or the operating system. For example the Accessibility API in the OS X operating system as described above provides information regarding content items associated with a user interface element. Since an open content item handle may not reliably determine whether a content item is being edited by a native application **255**, these behaviors are used by presence management module **204** to determine presence relating to editing or modifying a content item as described further below.

Native applications **255** may typically be executed on device **100** independently from one another, and may permit communication between the applications and other applications or processes executing on device **100**. Native applications **255** typically provide information to processes using application programming interfaces (APIs), which permit applications to request information from the executing process. For example, native applications **255** may present an API permitting a request for user interface elements controlled by the application, or to indicate the title of a user interface element, or to request a path in a file system associated with a content item opened by the native application **255**. Similarly, operating system **245** may provide similar APIs to requesting processes, such as requesting information about a process that controls a particular user interface element.

Client application **200** manages access to content management system **110**. Client application **200** includes user interface module **202** that generates an interface to the content accessed by client application **200**, as variously illustrated herein, and is one means for performing this function. The generated interface is provided to the user by display **220**. Client application **200** may store content accessed from a content storage at content management system **110** in local content data store **208**. While represented here as within client application **200**, local content data store **208** may be stored with other data for client device **100** in non-volatile storage. When local content data store **208** is stored this way, the content is available to the user and other applications or modules, such as native application **255**, when client application **200** is not in communication with content management system **110**. Content access module **206** manages updates to local content data store **208** and uses synchronization logic to communicate with content management system **110** to synchronize content modified by client device **100** with content maintained on content management system **110**. One example of such synchronization is provided in U.S. patent application Ser. No. 14/040,584, filed Sep. 27, 2013 and is hereby incorporated by reference in its entirety. Client application **200** may take various forms, such as a stand-alone application, an application plug-in, or a browser extension.

Content management system **110** may provide additional data for synchronizing content items, such as information designating that a content item has been deleted, or that the device **100** may be viewing or editing an outdated version of a content item.

Interaction management module **204** obtains and manages interaction information relating to a user's synchronized content items. As described above, the interaction management module **204** is typically a distinct module from the native applications **255** being monitored by interaction management module **204** for presence information and executes as a separate process. Interaction management module **204** sends interaction information determined about a content item synchronized to device **100** to content management system **110**. Interaction management module **204** also receives interaction information relating to other users from content management system **110** for display to the user. As described further below, in one embodiment the interaction management module **204** displays presence information relating to other users by attaching a presence indicator to a user interface element associated with a synchronized content item. Further interaction information is also displayed with the presence indicator.

To determine many types of interaction information, interaction management module **204** receives interaction information through user interface elements, as further described below. To determine presence information related to a synchronized content item, interaction management module **204** monitors user interface elements associated with native applications **255**. Interaction management module **204** can monitor all user interface elements, or alternatively monitor just certain user interface elements after the user interface element is associated with a content item. Monitored presence data store **210** includes information maintained by interaction management module **204** to indi-

cate that particular user interface elements are monitored to determine actions relating to a synchronized content item.

While shown here as a part of client application **200**, in various implementations the content access module **206** and interaction management module **204** are separated into distinct modules for performing their respective functions. Similarly, various modules and data stores are described separately throughout this disclosure for convenience and in various implementations may be combined or further separated into separate components as desired.

FIG. **4** shows an example process for determining presence information associated with a content item according to one embodiment. This process is typically performed by interaction management module **204**. Where the user interface elements are monitored only after being associated with a content item, interaction management module **204** uses events indicating that a content item is being opened by an application or user interface element to determine whether to monitor a user interface element. This is one example of an event that may associate a content item with a user interface element to initiate monitoring of the user interface element, termed a monitoring event. In other embodiments, a selection of user interface elements to monitor is determined in another way, or all user interface elements are monitored, in which case the interaction management module **204** may not use monitoring events.

If enabled by operating system **245**, the interaction management module **204** may register with operating system **245** to receive monitoring events for specific applications. In these embodiments, operating system **245** notifies interaction management module **204** when a request to open a content item is received by operating system **245**. In this embodiment, interaction management module **204** receives a monitoring event **400** that indicates a window or other user interface element is interacting with a content item, which may be a synchronized content item (i.e., the process is interacting with the content item in a particular user interface element). The monitoring event designates at least a user interface element that triggered the monitoring event. In other embodiments, interaction management module **204** monitors events associated with user interface elements from time-to-time (e.g., five minute intervals) and queries whether the user interface elements are associated with any open content items. According to operating system **245** and native application **255** configuration, this query may be directed to operating system **245** or native application **255**. When a user interface element is associated with a newly opened content item, that newly opened content item is treated as a monitoring event to determine whether the newly opened content item is a content item synchronized with content management system **110** and that presence information should be determined for the newly opened content item.

When the monitoring event is received, interaction management module **204** determines **410** which process is responsible for the user interface element associated with the monitoring event. Interaction management module **204** typically determines the process by requesting the process ID associated with the user interface element from operating system **245**. In some embodiments, the interaction management module **204** identifies a process by requesting an identification of the process from the user interface element itself.

To confirm that the process and user interface element are correctly associated with one another and that the user interface element is still active, interaction management module **204** may also request from the process the identity of the currently active user interface element. The interaction management module **204** confirms that the currently active user interface element received from the process matches the user interface element associated with the monitoring event.

Using the process identifier, interaction management module **204** requests **420** any open content item from the process to obtain an associated directory path for the content item. The interaction management module **204** may designate the user interface element associated with the monitoring event with the request for the open content item's path. The interaction management module **204** requests the open item from the process or operating system using an interface available interface to the process or operating system. As one example, in the OS X operating system, the accessibility API may be used to access information relating to a content item and content item path for a user interface element, as known in the art. Using the content item path provided by the process, the interaction management module **204** determines whether the opened content item path corresponds to any synchronized content items. If so, interaction management module **204** determines that the content item accessed by the process is a content item synchronized to content management system **110** and associates that process and user interface element with the content item. In other embodiments, other methods may be used to determine whether a content item accessed by the process is a synchronized content item.

If the content item is synchronized **430** to content management system **110**, interaction management module **204** stores information relating to the content item, process, and user interface element, to monitor **440** the user interface element for events. When the content item associated with the monitoring event is not synchronized, the process ends and the content item is not monitored. This monitoring information is stored in monitored presence data store **210**. To monitor and subsequently receive presence events related to the user interface element, interaction management module **204** registers to receive events associated with the user interface element. The registration process by the interaction management module **204** varies according to the configuration of device **100**. Typically the interaction management module **204** registers a request to receive presence events from operating system **245** or from the applicable process or user interface element. While the monitoring events determined whether a user interface element or process interacting with a synchronized content item, presence events are events that may indicate a change in state of a user's presence relating to the user interface element or process associated with a content item. Example presence events include a change in focus of a user interface element, closing a user interface element, closing a content item, opening a content item, and so forth based on the types of presence recognized by the interaction management module **204**. In various configurations, the presence events used by interaction management module **204** depend on the events operating system **245** and native application **255** make available for receipt by interaction management module **204**.

The presence events are used to determine presence information associated with the content item to which the presence event relates. For example, a presence event indicating that a user interface element that is associated with a content item has the focus will indicate that the user is viewing the content item, and hence the presence information for that content item indicates that state. Likewise, a presence event indicating that a user interface element unrelated to a content item has gained focus indicates that

the content item associated with a previously focused user interface element has lost focus, and thus indicates that user is no longer be viewing the content item. Thus, presence information provides a level of semantic interpretation of the underlying presence event itself.

In addition to receiving presence events that the interaction management module **204** registered for, presence events may also be initiated by interaction management module **204** to confirm that presence information has not changed for a monitored user interface element. These presence events may be initiated if a threshold amount of time passed since the last presence event for a particular user interface element or process, or at particular intervals, e.g., every five minutes.

In addition to registering for events, interaction management module **204** may receive interaction events in other ways. In one embodiment, users may expressly indicate interaction information through a user interface element. The user interface element can be configured to allow the user to indicate, for example, that a user intends to revise a content item, to indicate that intent to other users who are editing or viewing the content item, for example by selection of a menu item or icon that represents the particular intent. The user interface element can also be configured to allow a user to indicate other intentions of the user, such as a user's intention to no longer view a content item, or to expressly indicate that a user is not or will not be present for a content item. Other users may use such "not present" intention to know that the content item is free for editing. User input interaction events may also include messages or chat features to be disseminated to other users associated with the content item, for example, to transmit a message to other users currently viewing the content item on other devices.

When a presence event is received **450**, interaction management module **204** determines whether any presence information has changed since the last presence event related to a monitored user interface element. For user-initiated interaction information, the interaction information may be the information provided by the user, for example the user's selection of a user interface element indicating that the user intends to modify a content item, or a user's chat message. For presence events, the interaction management module **204** queries the monitored process to determine the status of the monitored user interface element. In particular, the interaction management module **204** queries the process to determine if the monitored user interface element is the active user interface element. When the monitored user interface element is the active user interface element, the content item is being viewed by the user.

In some embodiments, in addition to detecting user presence with respect to a content item, interaction management module **204** also determines whether the content item is being or has been modified by the user. This further aspect enables presence information to be reported more granularly, for example with an indication that a user has a presence with respect to the content item as an editor rather than as a viewer. As the particular actions performed by applications when a content item is being modified may vary as described above, detecting one of these actions by interaction management module **204** indicates that the process has edited the content item. For example, according to the type of actions expected by the process when the content item is edited, interaction management module **204** may query the process to determine if the process indicates the content item has been flagged as modified, if the title information of the user interface element has changed, if a temporary file has been saved or cached, or any other data that suggests the content item has been modified. Interaction management module **204** may also query the operating system to determine if a content item has been saved that matches a temporary content item format, for example a content item with a filename similar to the content item, but with a tilde or other specialized variation of the filename. Such modifications indicate that the presence information associated with the content item should reflect that the user is editing the content item.

After determining **460** the presence information, any new presence information for a user interface element may be stored as monitored presence data store **250**. This presence information in one embodiment is stored on a user interface element-by-user interface element basis, such that multiple user interface elements by one process may be associated with the same content item, and have presence information individually managed. In one embodiment, presence information may change based on the current presence status. For example, when the presence information for a content item reflects that the content item is being edited, in one embodiment the presence for the content item in a user interface element is not changed when a user changes focus to another user interface element. Instead, the edited status is maintained with respect to that user interface element until a presence event indicates the user interface user interface element is closed. In another embodiment, since editing has the potential to introduce modifications to the content item, in one embodiment the presence information for an edited document is not changed until the interaction management module **204** receives a notification that modifications to the content item are either committed or the modifications are discarded.

A content item with presence information indicating it is being viewed may have that status change when the user interface element loses focus, or within a threshold period of time of losing focus. This may be the case even if the user interface element associated with the content item is still open. In one embodiment, "viewed" presence information indicates whether a content item is associated with an active user interface element. In one embodiment, "viewed" presence information is retained until the user interface element is not active (or has lost focus) for longer than a threshold amount of time. In one embodiment, the content item is considered "viewed" while the content item is open by an application.

When there is a change to the interaction information, interaction management module **204** sends the interaction information to content management system **110**. In one embodiment, the sent presence information includes an identifier of the content item, the process id, the user interface element id, and the presence status.

FIGS. **5**A-**5**D show example user interfaces displaying interaction information. Various user interface elements are similar to those depicted in FIGS. **3**A and **3**B. In FIG. **5**A, the example user interface displays presence information. To display presence information, interaction management module **204** provides presence indicator(s) **500** along a boundary or border of the window associated with the content item. In this example, interaction management module **204** received presence information indicating that user Jeff is editing content item **2** and four users are viewing content item **2**. Interaction management module **204** displays presence indicator **500**A to indicate that Jeff is editing the document and presence indicator **500**B indicating users are viewing the document. In one embodiment, presence indicator **500** displays an icon or picture of a user associated with the presence indicator. The icon or picture may be obtained by

interaction management module **204** from contact directory **240**, or may be received from content management system **110**. The individual user displayed by the presence indicator may vary depending on the embodiment, and may display, for example, the first user having a particular presence (e.g., the first viewer) or the user who most recently opened the content item.

Presence indicator **500** is displayed along with the window associated with the content item, and in one embodiment interaction management module **204** tracks the location of the window and displays presence indicator **500** adjacent to or near the window, for example alongside a border or boundary of the window. A supplemental presence indicator **510** may appear when a user hovers over presence indicator **500** to provide further information or interfaces for the user. In the example shown in FIG. **5**A, supplemental presence indicator **510** describes presence indicator **500**A, specifically that Jeff is associated with a presence of editing content item **2**. Supplemental presence indicator **510** may also appear when a presence changes, to indicate a new user is viewing or editing the document, for example.

The presence indicator **500** and supplemental presence indicator **510** may be located on any convenient area of display **220**. In one embodiment the presence indicator is displayed proximal the associated user interface element of the content item so as to visually indicate to the user the relationship between the presence indicator and the specific content item. In addition, the display of the presence indicator along a boundary or border of the window increases the likelihood that the user will notice the presence indicator. In one embodiment, the presence indicators **500** are displayed on or alongside a vertical edge of the window containing the content item (e.g., right edge as shown FIG. **5**A).

Alternatively, the presence indicator may be shown in a separate area of the display, such as a taskbar, or tray icon or may be a separate user interface element that does not interact with the user interface element of the content item. Though shown here as a single presence indicator **500** for each type of presence (editing or viewing), any number of presence indicators **500** may be shown related to the content item. For example, a circular indicator may include a count of users viewing the content item and another circular indicator may include a count of users editing the content item. Alternatively, where multiple presence indicators **500** are to be displayed, they may be ordered from top to bottom, where the ordering can be most recent to least recent, or highest priority to lowest priority, or a combination thereof (e.g., ordered by priority, and for presence indicators of the same priority, ordered by recency). The presence indicator, as shown, may also indicate an icon or picture associated with the other user. The indicators may also be color-coded to indicate the risk that a user will affect edits by other users. For example, the presence indicator may be red (or the presence indicator may turn red) when another user is editing, indicating to the user to coordinate any desired modifications with that user's changes. Likewise, the editing indicator may be yellow when other users are viewing the content item, and green when the current user is the only user viewing the content item.

FIG. **5**B shows a user interface through which a user may enter interaction information to the interaction management module **204**. This interface includes presence indicators **500**, in addition to further user interface elements. This interface may be presented in lieu of the example of FIG. **5**A, or may be presented as a supplemental presence indicator providing additional data regarding the content item.

This user interface provides notification element **520**, permitting a user to indicate the user's desire to be notified when a presence change occurs related to another user. Upon selecting notification element **520**, the user may further select specific notification options, for example to be notified when a particular user's presence changes (e.g., Jeff is no longer editing), that all users have stopped a particular presence (e.g., no one is editing), or that no users have an active presence (e.g., that no one else is accessing content item **2**). The user's notification preference is sent to content management system **110** or the applicable users to request notification when the requested presence changes.

This interface also provides a chat interface **530** for users to communicate with other users present in the content item. The chat interface **530** permits users to enter and receive messages to other users. Presence changes may also be indicated in the chat interface **530**, for example that Amanda is now viewing the content item. The chat interface **530** may permit users to specifically discuss information relating to that content item, such as when a user expects to finish editing the item. These chat messages are received by interaction management module **204** as interaction information and sent to other clients synchronized to the content item. This permits users to chat directly about a content item, even if the native application provides no chat functionality.

Notes user interface **540** permits a user to retrieve and enter notes stored in association with the content item. When notes interface **540** is selected, interaction management module **204** requests any notes or other persistent information from content management system **110** relating to the content item and displays any such notes to the user along with an interface for entering additional notes to be sent to other synchronized devices and content management system **110**. Like the chat window, this permits additional notes to be entered for a content item and application providing the user interface which may not natively provide for any note functionality. The notes element **540** may also be used when no other users are present within the content item and may be used to leave messages for other users about a content item.

Content data element **550** permits a user to request additional data about the content item, such as any related metadata, user permissions, and so forth. This permits a user to request details of the content item from the content management system directly from an interface near the user interface element associated with the content item.

Similarly, versions element **560** indicates a request for version information. Interaction management module **204** transmits the request for version information to content management system **110** as interaction information related to the content item. In response, content management system **110** identifies relevant version information for the content item. In one embodiment, a prior version is compared side-by-side with the version of the content item displayed in the user interface element. In another embodiment, the prior version is compared in-line in the user interface element.

FIG. **5**C shows a user interface in which a user is notified about a change in presence related to another content item. In this case, a user had previously selected to be notified when user Jeff finished editing content item **2**, using notification element **520**. After requesting notification, the user proceeded to close window **310**B relating to content item **2**, and is presently viewing content item **1**. In this embodiment, presence indicator **500**D indicates that the user is the only user with presence relating to this content item. This may be expressly indicated ("No one else is viewing this file") as

shown, or may be provided if a user interacts with presence indicator **500**D. Interaction management module **204** receives a notification that Jeff has finished editing content item **2**. The notification **570** is displayed to the user. This notification is provided though the user may be interacting with another application or another content item than the content item to which the notification relates.

FIG. **5**D shows a user interface in which a user receives a notification that a content item has been modified. In this example, a new version of content item **2** viewed by the user has been saved by another user, Jeff. Interaction management module **204** receives a message from content management system **110** indicating that content item **2** has been modified and a new version is available. Typically, the user's edits may be lost if the user attempted to synchronize the user's edits to content management system **110** because the user's edits would be out of date with the version maintained by content management system **110**. Using the presence information indicating the user is editing the content item, the user can be provided choices to assist in incorporating any edits from Jeff into the user's edits. A version notification **580** is presented to the user, along with choices to make in response to the new version. In this example, the user may elect to load the new version, discard the user's edits, or save the user's edits as a new file. The particular choices available to a user may be provided by content management system **110** based on content management system **110**'s record of the device's presence **100**, as further described below. In another embodiment, interaction management module **204** determines options for display to the user.

In embodiments where interaction management module **204** is a separate application from the native application **255** of window **310**B, presence indicators **500** are provided by interaction management module **204**. Thus, interaction management module **204** monitors presence information associated with the application, and displays presence information relating to other users for a content item associated with the application. When other windows are activated, the same interaction management module **204** displays presence information relating to the activated window.

To display the presence indicator(s), interaction management module **204** receives the presence information for other devices, typically from the content management system **110**. Interaction management module **204** determines that a synchronized content item is in use by a window or process, and that the window is being displayed to the user. For example, interaction management module **204** may receive a presence event from the window or process being monitored as described above. When the presence event indicates that the window has the focus, interaction management module **204** adds the presence indicator to the display. Interaction management module **204** in one embodiment also tracks movement of the window in the desktop and moves the presence indicators to maintain the location of the presence indicators relative to the window.

In one embodiment, to add the presence indicator to the display, interaction management module **204** determines the location of the focused window and its boundaries, and adds the presence indicator adjacent to the window boundary. When interaction management module **204** identifies that the focused window has changed, the presence indicator for that content item may also be removed until that window is focused again.

FIG. **6** shows components of content management system **110** of FIG. **1**, according to one embodiment. In one configuration, components described below with reference to content management system **110** are incorporated into devices **100** that share and synchronize content items without management by content management system **110**. These devices **100** may synchronize content and share interaction information over network **120** or via a direct connection as described above. In this configuration, devices **100** may incorporate functionality of synchronization module **612**, conflict management module **614**, interaction synchronization module **616**, and other modules and data stores for incorporating functionality described below as provided by content management system **110**. Accordingly, devices **100** in this configuration operate in a peer-to-peer configuration and may do so without content management system **110** or network **120**.

When using content management system **110**, to facilitate the various content management services, a user can create an account with content management system **110**. The account information can be maintained in user account database **618**, and is one means for performing this function. User account database **618** can store profile information for registered users. In some cases, the only personal information in the user profile can be a username and/or email address. However, content management system **110** can also be configured to accept additional user information, such as password recovery information, demographics information, payment information, and other details. Each user is associated with an identifier, such as a userID or a user name.

User account database **618** can also include account management information, such as account type, e.g., free or paid; usage information for each user, e.g., file edit history; maximum storage space authorized; storage space used; content storage locations; security settings; personal configuration settings; content sharing data; etc. Account management module **604** can be configured to update and/or obtain user account details in user account database **618**. Account management module **604** can be configured to interact with any number of other modules in content management system **110**.

An account can be associated with multiple devices **100**, and content items can be stored in association with an account. The stored content can also include folders of various types with different behaviors, or other content item grouping methods. For example, an account can include a public folder that is accessible to any user. The public folder can be assigned a web-accessible address. A link to the web-accessible address can be used to access the contents of the public folder. In another example, an account can include a photo folder that is intended for photo content items and that provides specific attributes and actions tailored for photos; an audio folder that provides the ability to play back audio file content items and perform other audio related actions; or other special purpose folders. An account can also include shared folders or group folders that are linked with and available to multiple user accounts. The permissions for multiple users may be different for a shared folder. In one embodiment, the account is a namespace that may be associated with several users, each of whom may be associated with permissions to interact with the namespace.

The content can be stored in content storage **620**, which is one means for performing this function. Content storage **620** can be a storage device, multiple storage devices, or a server. Alternatively, content storage **620** can be a cloud storage provider or network storage accessible via one or more communications networks. In one configuration, content management system **110** stores the content items in the same organizational structure as they appear on the device. However, content management system **110** can store the content items in its own order, arrangement, or hierarchy.

Content storage **620** can also store metadata describing content items, content item types, and the relationship of content items to various accounts, folders, or groups. The metadata for a content item can be stored as part of the content item or can be stored separately. In one configuration, each content item stored in content storage **620** can be assigned a system-wide unique identifier.

Content storage **620** can decrease the amount of storage space required by identifying duplicate content items or duplicate segments of content items. In one embodiment, for example, a content item may be shared among different users by including identifiers of the users within ownership metadata of the content item (e.g., an ownership list), while storing only a single copy of the content item and using pointers or other mechanisms to link duplicates with the single copy. Similarly, content storage **620** stores content items using a version control mechanism that tracks changes to content items, different versions of content items (such as a diverging version tree), and a change history. The change history includes a set of changes that, when applied to the original content item version, produces the changed content item version.

Content management system **110** automatically synchronizes content items from one or more devices, using synchronization module **612**, which is one means for performing this function. The synchronization is platform-agnostic. That is, the content items are synchronized across multiple devices **100** of varying type, capabilities, operating systems, etc. For example, client application **200** synchronizes, via synchronization module **612** at content management system **110**, content in the file system of device **100** with the content items in an associated user account on system **110**. Client application **200** synchronizes any changes to content items in a designated folder and its sub-folders with the synchronization module **612**. Such changes include new, deleted, modified, copied, or moved files or folders. Synchronization module **612** also provides any changes to content associated with device **100** to client application **200**. This synchronizes the local content at device **100** with the content items at content management system **110**.

Conflict management module **614** determines whether there are any discrepancies between versions of a content item located at different devices **100**. For example, when a content item is modified at one device and a second device, differing versions of the content item may exist at each device. Synchronization module **612** determines such versioning conflicts, for example by identifying the modification time of the content item modifications. Conflict management module **614** resolves the conflict between versions by any suitable means, such as by merging the versions, or by notifying the device of the later-submitted version.

A user can also view or manipulate content via a web interface generated by user interface module **602**. For example, the user can navigate in web browser **250** to a web address provided by content management system **110**. Changes or updates to content in content storage **620** made through the web interface, such as uploading a new version of a file, are synchronized back to other devices **100** associated with the user's account. Multiple devices **100** may be associated with a single account and files in the account are synchronized between each of the multiple devices **100**.

Content management system **110** includes communications interface **600** for interfacing with various devices **100**, and with other content and/or service providers via an Application Programming Interface (API), which is one means for performing this function. Certain software applications access content storage **620** via an API on behalf of a user. For example, a software package, such as an app on a smartphone or tablet computing device, can programmatically make calls directly to content management system **110**, when a user provides credentials, to read, write, create, delete, share, or otherwise manipulate content. Similarly, the API can allow users to access all or part of content storage **620** through a web site.

Content management system **110** can also include authenticator module **606**, which verifies user credentials, security tokens, API calls, specific devices, etc., to determine whether access to requested content items is authorized, and is one means for performing this function. Authenticator module **406** can generate one-time use authentication tokens for a user account. Authenticator module **406** assigns an expiration period or date to each authentication token. In addition to sending the authentication tokens to requesting devices, authenticator module **606** can store generated authentication tokens in authentication token database **622**. Upon receiving a request to validate an authentication token, authenticator module **606** checks authentication token database **622** for a matching authentication token assigned to the user. Once the authenticator module **606** identifies a matching authentication token, authenticator module **606** determines if the matching authentication token is still valid. For example, authenticator module **606** verifies that the authentication token has not expired or was not marked as used or invalid. After validating an authentication token, authenticator module **606** may invalidate the matching authentication token, such as a single-use token. For example, authenticator module **606** can mark the matching authentication token as used or invalid, or delete the matching authentication token from authentication token database **622**.

Content management system **110** includes a sharing module **610** for sharing content publicly or privately. Sharing content publicly can include making the content item accessible from any computing device in network communication with content management system **110**. Sharing content privately can include linking a content item in content storage **620** with two or more user accounts so that each user account has access to the content item. The content can also be shared across varying types of user accounts.

In some embodiments, content management system **110** includes a content management module **608** for maintaining a content directory that identifies the location of each content item in content storage **620**, and allows client applications to request access to content items in the storage **620**, and which is one means for performing this function. A content entry in the content directory can also include a content pointer that identifies the location of the content item in content storage **620**. For example, the content entry can include a content pointer designating the storage address of the content item in memory. In some embodiments, the content entry includes multiple content pointers that point to multiple locations, each of which contains a portion of the content item.

In addition to a content path and content pointer, a content entry in some configurations also includes a user account identifier that identifies the user account that has access to the content item. In some embodiments, multiple user account identifiers can be associated with a single content entry indicating that the content item has shared access by the multiple user accounts.

To share a content item privately, sharing module **610** adds a user account identifier to the content entry associated with the content item, thus granting the added user account access to the content item. Sharing module **610** can also be

configured to remove user account identifiers from a content entry to restrict a user account's access to the content item.

To share content publicly, sharing module **610** generates a custom network address, such as a URL, which allows any web browser to access the content in content management system **110** without any authentication. The sharing module **610** includes content identification data in the generated URL, which can later be used by content management system **110** to properly identify and return the requested content item. For example, sharing module **610** can be configured to include the user account identifier and the content path in the generated URL. The content identification data included in the URL can be transmitted to content management system **110** by a device to access the content item. In addition to generating the URL, sharing module **610** can also be configured to record that a URL to the content item has been created. In some embodiments, the content entry associated with a content item can include a URL flag indicating whether a URL to the content item has been created.

Content management system **110** may be implemented using a single computer, or a network of computers, including cloud-based computer implementations. For the purposes of this disclosure, a computer is device having one or more processors, memory, storage devices, and networking resources. The computers are preferably server class computers including one or more high-performance CPUs and 1G or more of main memory, as well as 500 Gb to 2 Tb of computer readable, persistent storage, and running an operating system such as LINUX or variants thereof. The operations of content management system **110** as described herein can be controlled through either hardware or through computer programs installed in computer storage and executed by the processors of such server to perform the functions described herein. These systems include other hardware elements necessary for the operations described here, including network interfaces and protocols, input devices for data entry, and output devices for display, printing, or other presentations of data, but which are not described herein. Similarly, conventional elements, such as firewalls, load balancers, failover servers, network management tools and so forth are not shown so as not to obscure the features of the system. Finally, the functions and operations of content management system **110** are sufficiently complex as to require implementation on a computer system, and cannot be performed in the human mind simply by mental steps.

Interaction synchronization module **616** manages synchronization of interaction information across devices **100**. Devices **100** provide interaction information to interaction synchronization module **616**. Interaction synchronization module **616** responds to interaction information based on the type of interaction information received. As interaction synchronization module **616** uses presence information to determine a response for certain types of interaction information, treatment of received presence information is described first, with treatment of additional types of interaction information described further below.

Interaction synchronization module **616** receives presence information from a device, stores it as part of a presence record in presence data store **624** and determines a user presence with respect to a content item. Each user may be associated with a user presence describing presence records associated with that user with respect to a content item, which may be without reference to any particular user device, process, or user interface element. While presence information may describe presence with respect to a particular user interface element or process, this presence associated with a user is termed a user presence. Example user presence includes editing, viewing, open, and not present. In this example, an "editing" user presence indicates the content item is associated with a user interface element that has modified the content item, a "viewing" user presence indicates the content item is associated with an active user interface element on a device **100**, while a "open" user presence indicates a user interface element is associated with the content item and has opened the content item, but has not yet closed the content item. Various embodiments may use more or fewer user presences. For example, one embodiment includes only "editing" "viewing" and "not present," in which case user interface elements that have opened the content item but are not the active user interface element may be treated as viewing or not presence, according to the configuration of the system.

FIG. **7** shows an ordering of user presence according to one embodiment. The ordering as shown in FIG. **7** describes the priority of interaction with a content item associated with each user presence. In this example, a user has a higher priority of interaction with a content item when the user is editing a content item relative to viewing the content item, and likewise viewing a content item in an active user interface element is a higher priority of interaction than a user interface element that has the content item open.

Since a user may open a content item across several devices and across several processes and user interface elements, the ordering is used to determine the highest priority of user interaction of that user. This permits the user presence to reflect the highest priority of user presence associated with the various possible devices and user interface elements associated with a user. This user presence is sent to devices **100** for display as shown in FIG. **5**A, and permits a single user presence indicating the highest priority of user interaction to be shown.

FIG. **8** shows a presence record in presence table **800** relating to a content item. Presence table **800** may be stored, for example, in presence data store **624**. Presence information received by interaction synchronization module **616** is converted into a presence record for storage in presence data store **624**. When presence records are stored in presence data store **624**, the presence records may be stored in key-value storage, for example, using an identifier of the content item as a key. Presence table **800** may be stored by presence data store **624**, or may conceptually represent information retrieved by interaction synchronization module **616** from presence data store **624**.

In this example, a presence record identifies a content item, a device id, a user id (for convenience represented here as a name), a user interface element id, a process id, and presence information. Each unique device id, user interface element id, and process id combination is associated with presence information. In this example, devices which are "not present" are not represented in presence table **800**. As shown by presence table **800**, individual users (Jim) and devices (e.g., device id 36abe87) may be associated with multiple presence records. The presence table in this embodiment describes information relating to individual processes and user interface elements operated by various users and devices as provided by interaction management module **204**.

When presence information is received by interaction synchronization module **616**, interaction synchronization module **616** accesses presence data store **624** to identify whether an existing presence record matches the device id, process id, and user interface element id of the received presence information. When no record matches this infor-

mation, a new presence record is generated. The "user ID" of the new presence record may be included in the received information, or interaction synchronization module **616** may query another module or data store to identify the user associated with the device ID. The new presence record is stored in presence data store **624**. When a matching presence record is identified in presence data store **624**, the existing presence record is updated based on the received information. When the received information indicates the content item is no longer associated with the user interface element (e.g., because the relevant window has closed), the presence record may be removed from presence data store **624**.

In this example, interaction synchronization module **616** also stores a presence record in presence table **800** reflecting a user's request to be notified when a particular presence event occurs. In various embodiments, a user's notification request may be stored in any suitable way. The notification request is received by interaction synchronization module **616** as interaction information input via notification element **520** and sent to interaction synchronization module **616** by the device receiving the notification request. The notification request may be stored with sufficient information describing the notification request, such as the requesting device and the user requesting the notification. The user may be used to identify and notify alternate devices if the user is no longer using the device that provided the request.

Interaction synchronization module **616** may generate user presence table **810** to maintain the current user presence relating to a content item. In other embodiments, the current user presence is stored without using user presence table **810**. User presence table **810** indicates the presence of a user reflecting the presence records associated with the user devices of the user. The user presence is used by devices **100**, for example, to display in presence indicators **500**. In this way, though a user may interact with a content item using several devices, processes, and user interface elements, only the presence reflecting the highest priority of interaction is distributed to other devices **100**.

To determine the user presence, the presence information from each presence record associated with a user is identified. Then the presence information is matched against the user presence ordering to determine the highest priority ordering. With user John as an example, John is associated with three user presences. The presence records indicate that John is viewing the content item in a focused window of one user device (i.e., Device 9d7523c in user interface element id 1), and in two user interface elements of device 36abe87, the content item is being edited in the first and "open" in the second. Applying the ordering shown in FIG. **7**, "editing" is the highest order presence associated with John. John's user presence is set to "editing" in user presence table **810**.

When a presence record is modified, the user presence is updated to reflect any changes in the user presence. For example, if John's "viewing" presence record is removed, John's user presence is not changed because an "editing" presence record remains. Likewise, if John's "editing" presence record changes to "open," John's presence status changes to "viewing" as the next-highest priority presence reflected in presence records associated with John. Note that the "viewing" presence record is associated with another device ID (9d7523c). Thus, John's presence is accurately reflected across several devices, processes, and user interface elements.

FIG. **9** shows a process of modifying a presence record and updating devices **100** of user presence according to an embodiment. This process may be performed in one embodiment by interaction synchronization module **616**. Presence information is received **900** by interaction synchronization module **616**. Interaction synchronization module **616** updates **910** presence records in presence table **800** by modifying existing presence records or adding a new presence record, as described above. Interaction synchronization module **616** next determines **920** the user presence of the user associated with the updated presence record and determines whether the user presence is changed by the updated presence record. For example, a change in user presence may be determined by determining a new user presence and comparing the new user presence to the user presence stored in user presence table **810**. If the user presence was not changed, in one embodiment, the process ends. If the user presence changes, the user presence is updated **930** in user presence table **810**.

In one embodiment, interaction synchronization module **616** determines whether a user interface element of a device has provided presence information within a threshold amount of time. When a user interface element has not provided presence information, interaction synchronization module **616** may request presence information from the device or consider the presence for the related user interface element to have ended ("not present"). Thus, interaction synchronization module **616** may "expect" a heartbeat relating to each user interface element that has provided recent presence information. When no presence information is received within the threshold amount of time, a "not present" presence may be generated for that user interface element.

Next, interaction synchronization module **616** identifies **940** devices to update regarding the changed user presence. In one embodiment, any device synchronized with the content item is updated. In another embodiment, the presence table **810** is consulted to determine devices with a presence associated with the content item. In addition, interaction synchronization module **616** determines whether any notifications are triggered by the change in user presence. Stored notifications are accessed and the conditions for the notification are compared to the changed presence related to the content item. Using identified devices, interaction synchronization module **616** sends **950** the updated presence information to the identified devices. In one configuration, the user presence is provided to the identified devices. In another configuration, additional information may also be provided. In this way, user presence information is managed by content management system **110** to distribute presence information only when it affects the user presence that may actually displayed to a user, e.g., in a presence indicator or responsive to a notification request.

Additional interaction information provided by devices **100** is also processed by interaction synchronization module **616**. Such interaction information includes a request to be notified when presence of a user changes, a request for version or metadata information, chat messages, and so forth as described with respect to FIG. **5**B. As described above, notification requests may be stored and used to determine devices to notify when presence status changes. When version or metadata is requested, interaction synchronization module **616** retrieves such version and metadata information from content storage **620**. Interaction synchronization module **616** may provide the version information and changes relative to various content item versions as described with respect to content storage **620**. When a chat message is received for a content item, interaction synchronization module **616** queries presence data store **624** to identify

device IDs associated with stored presence records for the content item and distributes the chat message to these identified devices.

When changes to a content item are made by a device, presence records may be used to intelligently and smoothly update individual user interface elements interacting with the content item. Such changes to a content item include moving the content item (e.g., to another folder or directory), deleting the content item, and committing a new version of the content item.

FIG. **10** shows a flowchart for a process that uses presence information to synchronize modifications to a content item. This process may be performed by interaction synchronization module **616** or by client application **200**, according to various embodiments. When a change to a content item is received **1000**, presence records for the affected content item are accessed **1010** to determine **1020** the presence information associated with each user interface element. Depending on the presence of the user interface element and the change to the content item, an action is sent to the client application for the user interface element to perform to reconcile the user interface element with the change to the content item.

When a user interface element is associated with an "open" presence **1030** (i.e., the user interface element has interacted with the content item, but is not the active user interface element in the display), the action sent to the process controlling the user interface element typically instructs the process to seamlessly integrate the content item change. For example, when a new version of the content item is committed, the process is instructed to close and automatically re-open the new, changed content item. When the content item is moved, the process may be instructed to close the content item and re-open the content item at the new location. When the content item is deleted, the process may be instructed to close the content item. When the presence record indicates the user interface element is viewing **1040** (e.g., presently the active user interface element on the device), a user may be notified prior to performing the action described for the "open" presence information. In addition, the user may be notified of the specific change to the content item and prompted to perform further actions. While a user may not mind if a content item is moved to another folder, a user may prefer an option to save the current version of a content item prior to synchronizing a deletion or modification of the content item. The action in this circumstance may include prompting the user, e.g., via a supplemental user interface, whether to continue with synchronizing the modification, to save the current content item, or to continue viewing the current content item without change.

When the presence record indicates the user interface element is associated with editing **1050** the content item (e.g., the user interface element is associated with an unsaved modification of the content item), the action may present the user with options to preserve the content item, including any edits the user has made or to preserve the unedited version of the content item. For example, when the change deletes the content item, the action includes presenting an option to save the modified content item. When the change moves the content item, the action may assist in synchronizing the edited content item with the new location of the content item. For example, the action may instruct the client application **200** to save any modifications of the content item.

When the change commits a new version of the content item and the user interface element is associated with modifying the content item (e.g., an editing presence), the action may notify the user of the new version and prompt the user to save the user's changes to another document (to avoid a versioning conflict with the existing content item) or to discard the user's changes. The action may also attempt to merge the newly committed version with the modifications made in the user interface element. Such merging may include identifying the committed changes to the content item and instructing the user device to attempt to incorporate the committed changes to the content item in the user interface element.

FIG. **11** shows a folder-level display of interaction information according to one embodiment. In the embodiment shown in FIG. **11**, user interface module **202** on device **100** displays a browser **1100** to the user for navigating synchronized content items. The synchronized content items may be associated with individual folders or shared folders as described above. Browser **1100** displays content items synchronized with content management system **110** and permits the user to navigate synchronized content items and organizations of such content items. In the example shown in FIG. **11**, browser **1100** shows a collection of content items for a shared folder. This view of a collection (e.g., a folder) is also termed a specific "organization" of content items displayed by browser **1100**. The organization of a shared folder may be hierarchical, and may include, for example, several organizational elements within the collection. Each organizational element is a content item that further organizes content items within the organization. For example, one type of organizational element is a folder, which may include one or more folders within the folder. Other organizational elements may be used, such as a collection, a set, a playlist, or any other suitable grouping for organizing and managing content items. A user may hierarchically navigate an organization of content items by selecting organizational elements and navigating the organizational structure. Browser **1100** displays interaction information relating to the content items viewed in the browser **1100**. In this way, as a user navigates shared and synchronized folders, the user may also view interactions with content items. As described further below, this permits a user to identify current and recent interactions with shared folders and content items.

In one embodiment, as interaction information is received by a client, the client associates the received interaction information with a shared folder and any related organizations and organizational elements. For example, a content item may be associated with folder A within a particular shared folder. The interaction information may be associated with the content item, the shared folder, and folder A. As shown in FIG. **11**, browser **1100** may display several content items including organizational elements. In this example, browser **1100** shows several folder organizational elements, as well as content items "Project Task List" and "Q1 Revenue Report."

The interaction information and other information about synchronized content items is analyzed to generate a status indicator (specifically one of status indicators **1110**A-F, and generally status indicator **1110**) for each content item, including organizational elements. The status indicator provides visual information to display interaction information or synchronization status for a content item or organizational element. In this example, status indicator **1110** may indicate that the associated item (content item or organizational element) is up to date, is currently receiving updates or modifications, or is associated with presence information. Status indicators associated with an organizational element may be determined by identifying content items and further

organizational elements contained by the organizational element, and displaying the status of the items contained by the organizational element.

In this example, status indicator **1110**A indicates that the folder "Tax Forms" is currently synchronized with content management system **110**. The folder "Tax forms" is an organizational element that is not shared with other users.

Status indicator **1110**B indicates that the folder "Music" is currently receiving modifications and synchronizing the status of the content items within this organizational element with content management system **110**.

Status indicator **1110**C indicates that the folder "Shared House Expenses" is currently synchronized with content management system **110**. This folder is shared with and synchronized by other users, as indicated by the designation "shared folder."

Status indicator **1110**D indicates that the folder "Quarterly Revenue Reports" is currently associated with presence information relating to eight other users, as indicated by "+8". In this embodiment, the number of users with presence information relating to an organizational element or content item is designated by a counter. The presence information relating to "Quarterly Revenue Reports" is associated with content items and other organizational items organized by this organizational item. For example, six users may be associated with presence information relating to content items in "quarterly revenue reports," while two users may be associated with a further organizational element within "quarterly revenue reports." That is, a further folder may exist within "quarterly revenue reports" within which the two users may be interacting with content items. Since the user is viewing "quarterly revenue reports," as an organizational element (and not as the organization displayed by the browser), the browser displays information consolidated interaction information associated with the content items within "quarterly revenue reports." The user may select the organizational element to browse to the organizational element itself and further identify the content items being viewed or otherwise interacted with (e.g., edited) by other users.

Similarly, status indicator **1110**E indicates that the folder "Efficiency Improvement Project" is currently associated with presence information relating to two other users as indicated by "+2".

Status indicator **1110**F indicates that the content item "Project task list" is associated with presence information for a user whose icon is displayed. In various methods of displaying such information, the user's icon may be displayed, or the status indicator may show a +1 or other non-personalized indicator. In this way, in the browser **1100**, a user may view interaction information relating to various content items and organizational elements associated with the view of the browser, permitting a user to see at-a-glance what items or organizational elements are being interacted with by other users.

In addition to status indicators **1110**, browser **1100** may also display a list of active users **1120** associated with the various status indicators **1110**. The user may interact with the list of active users **1120** in conjunction with filter icon **1150** to select individual users to view. For example, a user sees that user "John" in active users list **1120** is active and associated with presence information. The user may select filter **1150** to select "John" and the browser, managed by user interface module **202**, filters the displayed interaction information to the information that relates only to "John."

In addition, a user may select a user in active users list **1120** (or otherwise) to identify specific content items that the user is interacting with. Specifically, the presence information associated with that user is identified and the resulting content items are displayed. This may permit a user to quickly identify which content item another user is active with, even if that user is active in a content item in a sub-sub-sub-folder or other organization element deep in an organizational hierarchy. Thus, the viewing user may quickly "find" where another user is active within the organization of content items.

The browser **1100** also displays an activity feed **1130** related to the view displayed to the user and the various content items and organizational elements associated with the organization currently displayed. The activity feed may display any interaction information as described with respect to a user's view of a content item discussed with respect to FIGS. **5**A-**5**D. For example, activity feed **1130**A displays when a user's presence changes (e.g., opening or closing a content item). Activity feed item **1130**B indicates that a user has saved a new version of a file, and prompts a user to select a response, as previously discussed with respect to FIG. **10**. In the browser **1100**, modifications to a content item may not be displayed in activity feed **1130** until the modifications have been successfully synchronized to local content data store **208**.

As shown by activity feed item **1130**C, chat messages relating to a specific organization item (e.g., the folder "Efficiency Improvement Project") or relating to a specific content item (e.g., content item "Project Goals.txt") are displayed in the activity feed **1130**. Similarly, changes in presence activities of a user with respect to a content item within an organizational item may also be displayed, as shown by activity feed items **1130**D and **1130**E.

Finally, the activity feed **1130** may also notify the user of various changes to metadata relating to a shared folder, content item, or organizational item. For example, modifications, deletions, or additions to a content item, or the addition **1130**F of a new users to a shared folder. Users may also enter chat messages or other interaction information in a chat element **1140**. This message is associated with the organizational view of the user, in this example with the "My Workspace" shared folder. This allows users to communicate about a content item or a folder or shared folder from the browser **1100** itself.

In this way, browser **1100** displays detailed presence and interaction information relating to various content items and organizational elements without requiring the user to enter a native application of a content item, and allows the user to view changes and data about a content item at a glance.

Accordingly, the presence associated with each user interface element is thus used to smooth the user experience when a content item is modified. Thus, a user may be notified of content item notifications prior to the user closing user interface elements, and reduces the likelihood that a user will be surprised if a content item changes or is deleted by a synchronization shortly after the user closes it.

The foregoing description of the embodiments of the invention has been presented for the purpose of illustration; it is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Persons skilled in the relevant art can appreciate that many modifications and variations are possible in light of the above disclosure.

Some portions of this description describe the embodiments of the invention in terms of algorithms and symbolic representations of operations on information. These algorithmic descriptions and representations are commonly used by those skilled in the data processing arts to convey the substance of their work effectively to others skilled in the

art. These operations, while described functionally, computationally, or logically, are understood to be implemented by computer programs or equivalent electrical circuits, microcode, or the like. Furthermore, it has also proven convenient at times, to refer to these arrangements of operations as modules, without loss of generality. The described operations and their associated modules may be embodied in software, firmware, hardware, or any combinations thereof.

Any of the steps, operations, or processes described herein may be performed or implemented with one or more hardware or software modules, alone or in combination with other devices. In one embodiment, a software module is implemented with a computer program product comprising a computer-readable medium containing computer program code, which can be executed by a computer processor for performing any or all of the steps, operations, or processes described.

Embodiments of the invention may also relate to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, and/or it may comprise a general-purpose computing device selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a non-transitory, tangible computer readable storage medium, or any type of media suitable for storing electronic instructions, which may be coupled to a computer system bus. Furthermore, any computing systems referred to in the specification may include a single processor or may be architectures employing multiple processor designs for increased computing capability.

Embodiments of the invention may also relate to a product that is produced by a computing process described herein. Such a product may comprise information resulting from a computing process, where the information is stored on a non-transitory, tangible computer readable storage medium and may include any embodiment of a computer program product or other data combination described herein.

Finally, the language used in the specification has been principally selected for readability and instructional purposes, and it may not have been selected to delineate or circumscribe the inventive subject matter. It is therefore intended that the scope of the invention be limited not by this detailed description, but rather by any claims that issue on an application based hereon. Accordingly, the disclosure of the embodiments of the invention is intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.

What is claimed is:

1. A method comprising:

receiving, by a first presence application executed by a processor on a first device, an event associated with a user interface element of a native application separate from the first presence application, wherein the first presence application is not integrated within the native application, and wherein the first presence application is configured to monitor a plurality of user interaction types with content items displayed within user interface elements of each of a plurality of different native applications executing as separate processes from the first presence application;

determining, by the first presence application, an indication that a content item synchronized with a second device and with a content management system has been opened by the native application based on the received event; and

in response to the indication that the content item has been opened by the native application and to determining that the user interface element of the native application is active based on the received event, sending, by the first presence application, presence information identifying the synchronized content item to the second device.

2. The method of claim **1**, further comprising:

receiving, by the first presence application, a second event indicating that the user interface element displaying the content item has been closed; and

responsive to receiving the second event indicating that the user interface element has been closed, sending by the first application, second presence information to the content management system, the second presence information identifying the synchronized content item and indicating a change in presence relative to the presence information.

3. The method of claim **1**, further comprising determining that the content item is being viewed in a focused window designated to receive user input and sending presence information describing the viewing to another device.

4. The method of claim **1**, further comprising determining that the content item is being edited by the native application and sending presence information describing the editing to another device.

5. The method of claim **4**, wherein determining that the content item is being edited by the native application is based on at least one of a received flag from the native application, a change in title of the user interface element, or a temporary file associated with the content item.

6. The method of claim **1**, wherein sending the presence information is further in response to the received event indicating that a status of the user interface element of the native application has changed from inactive to active.

7. The method of claim **1**, wherein the presence information is further sent to the content management system.

8. The method of claim **1**, further comprising

registering, by the first presence application, with an operating system executing on the first device to receive events associated with the user interface element associated with the native application executing on the first device; and

wherein the received event associated with the user interface element of the native application is received from the operating system operating on the first device.

9. The method of claim **1**, wherein the first presence application is not an add-in for the native application.

10. The method of claim **1**, wherein the first presence application is configured to simultaneously monitor the user interface elements of each of the plurality of different native applications.

11. A computer program product comprising a non-transitory computer-readable storage medium containing computer program code configured to cause a processor to perform steps for a first presence application comprising:

receiving a presence event indicating a change in state of a user's presence associated with a user interface element of a native application that is separate from the first presence application and that is interacting with a content item synchronized to a content management system and to one or more other devices, wherein the first presence application is not integrated within the native application, and wherein the first presence application is configured to monitor a plurality of user interaction types with content items displayed within user interface elements of each of a plurality of different native applications executing as separate processes from the first presence application;

determining whether the presence event indicates that the content item synchronized to the content management system has been opened by the native application; and

in response to the indication that the content item has been opened by the native application and to determining that the user interface element of the native application is active based the received presence event, send the change in state of a user's presence to the content management system.

12. The computer program product of claim **11**, wherein the computer program code is further configured to cause the processor to:

receive, by the first presence application, a content item open event indicating a request to open a second content item synchronized to the content management system by the native application;

monitoring the user interface element of the native application for events that may reflect a change in user presence in regards to the second content item.

13. The computer program product of claim **11**, wherein the presence event is an opening of the user interface element.

14. The computer program product of claim **11**, wherein the presence event is a change in focus of the user interface element.

15. The computer program product of claim **11**, wherein the presence event is a closing of the user interface element.

16. The computer program product of claim **11**, wherein the presence event comprises an editing of the content item.

17. The computer program product of claim **16**, wherein determining that the content item has been edited comprises identifying at least one of a modification flag, change in user interface element title, or temporary file associated with the content item.

18. A system comprising:

a processor configured to execute instructions;

a non-transitory, non-volatile storage medium containing instructions of a first presence application, which when executed by the processor cause the processor to perform the steps of:

receiving, by the first presence application, an event associated with a user interface element of a native application separate from the first presence application, wherein the first presence application is not integrated within the native application, and wherein the first presence application is configured to monitor a plurality of user interaction types with content items displayed within user interface elements of each of a plurality of different native applications executing as separate processes from the first presence application;

determining, by the first presence application, that a content item synchronized with a second device and with a content management system has been opened by the native application; and

in response to the content item being opened and to determining that the user interface element of the native application is active based on the received event, sending presence information identifying the content item to the second device.

19. The system of claim **18**, wherein the presence information indicates that the user interface element is a focused window designated to receive user input.

20. The system of claim **18**, wherein the presence information indicates that the content item is being edited by the native application.

21. The system of claim **20**, wherein determining the content item is being edited is based on a received flag from the process or a change in title of the user interface element.

22. The system of claim **18**, wherein the received event indicates that a status of the user interface element of the native application changed from inactive to active dicating an active user interface window changed to the user interface element.

23. The system of claim **18**, wherein the presence information is additional sent to the content management system.

* * * * *